AD-A106 769   AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH   F/G 13/2
A MODEL FOR SOLVING MULTIPERIOD MULTIRESERVOIR WATER RESOURCES --ETC(U)
MAY 81   D D COCHARD
UNCLASSIFIED   AFIT-CI-81-48D   NL

1 OF 3
AD
A106769

UNCLASS

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: 81-48D
2. GOVT ACCESSION NO.: AD-A106769
3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): A Model For Solving Multiperiod Multireservoir Water Resources Problems With Stochastic Inflows
5. TYPE OF REPORT & PERIOD COVERED: THESIS/DISSERTATION
6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): Douglas Dewitt Cochard
8. CONTRACT OR GRANT NUMBER(s):
9. PERFORMING ORGANIZATION NAME AND ADDRESS: AFIT STUDENT AT: Univ of Texas at Austin
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:
11. CONTROLLING OFFICE NAME AND ADDRESS: AFIT/NR
WPAFB OH 45433
12. REPORT DATE: May 1981
13. NUMBER OF PAGES: 268
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):
15. SECURITY CLASS. (of this report): UNCLASS
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:
16. DISTRIBUTION STATEMENT (of this Report): APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report):

16 OCT 1981

Fredric C. Lynch

18. SUPPLEMENTARY NOTES: APPROVED FOR PUBLIC RELEASE: IAW AFR 190-17

FREDRIC C. LYNCH, Major, USAF
Director of Public Affairs
Air Force Institute of Technology (ATC)
Wright-Patterson AFB, OH 45433

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

ATTACHED

81 10 27 257

UNCLASS

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

A MODEL FOR SOLVING MULTIPERIOD MULTIRESERVOIR WATER RESOURCES PROBLEMS WITH STOCHASTIC INFLOWS

Publication No._____

Douglas DeWitt Cochard, PhD.
The University of Texas at Austin, 1981

Supervising Professor: Paul A. Jensen

The model developed solves the multiperiod multireservoir water resources problem with stochastic inflows. Of unique importance is the development of a generalized network model which solves nonlinear nonseparable quadratic problems. Quadratic functions are used to measure the future value of water to the system. The nonseparable form stems from the realization that interaction exists between the benefits to be gained from a multireservoir system. Historically this interactive nature has been ignored due to the computational difficulty of measuring and solving such relationships. Also developed is a stochastic dynamic programming approach which utilizes the results of the network optimization as data for a least squares regression analysis. A quadratic function is fit to this data and is used to represent the future value of water to the system for the next period in the dynamic programming approach. This functional representation of the future value of water replaces the standard discrete matrix representation of dynamic programming and greatly reduces the dimensionality problems associated with the dynamic programming approach. In the end, this work represents a rare combination of generalized-nonlinear network flow programming, stochastic dynamic programming and regression analysis. Example problems are included along with an application to a four reservoir model of the Guadalupe River Basin in Texas.

81-48D

AFIT RESEARCH ASSESSMENT

The purpose of this questionnaire is to ascertain the value and/or contribution of research accomplished by students or faculty of the Air Force Institute of Technology (ATC). It would be greatly appreciated if you would complete the following questionnaire and return it to:

AFIT/NR
Wright-Patterson AFB OH 45433

RESEARCH TITLE: A Model For Solving Multiperiod Multireservoir Water Resources Problems With Stochastic Inflows

AUTHOR: Douglas Dewitt Cochard

RESEARCH ASSESSMENT QUESTIONS:

1. Did this research contribute to a current Air Force project?

( ) a. YES ( ) b. NO

2. Do you believe this research topic is significant enough that it would have been researched (or contracted) by your organization or another agency if AFIT had not?

( ) a. YES ( ) b. NO

3. The benefits of AFIT research can often be expressed by the equivalent value that your agency achieved/received by virtue of AFIT performing the research. Can you estimate what this research would have cost if it had been accomplished under contract or if it had been done in-house in terms of manpower and/or dollars?

( ) a. MAN-YEARS ________ ( ) b. $________

4. Often it is not possible to attach equivalent dollar values to research, although the results of the research may, in fact, be important. Whether or not you were able to establish an equivalent value for this research (3. above), what is your estimate of its significance?

( ) a. HIGHLY SIGNIFICANT ( ) b. SIGNIFICANT ( ) c. SLIGHTLY SIGNIFICANT ( ) d. OF NO SIGNIFICANCE

5. AFIT welcomes any further comments you may have on the above questions, or any additional details concerning the current application, future potential, or other value of this research. Please use the bottom part of this questionnaire for your statement(s).

NAME GRADE POSITION

ORGANIZATION LOCATION

STATEMENT(s):

FOLD DOWN ON OUTSIDE - SEAL WITH TAPE

AFIT/NR
WRIGHT-PATTERSON AFB OH 45433

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 73236 WASHINGTON D.C.

POSTAGE WILL BE PAID BY ADDRESSEE

AFIT/ DAA
Wright-Patterson AFB OH 45433

FOLD IN

81-48D

A Model For Solving Multiperiod Multireservoir Water Resources Problems With Stochastic Inflows

by

DOUGLAS DEWITT COCHARD, B.S.,M.S.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fullfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May, 1981

A MODEL FOR SOLVING MULTIPERIOD MULTIRESERVOIR WATER

RESOURCES PROBLEMS WITH STOCHASTIC INFLOWS

Approved By Supervisory Committee

Paul A. Jensen

Leon Lasdon

Larry W. Mays

J. Wesley Barnes

ACKNOWLEDGEMENTS

The author gratefully acknowledges all members of his Supervisory Committee for their assistance. Special thanks are due for Dr. Paul A. Jensen, the committee chairman, who worked closely with me. I would also like to thank Dr. James Wilson for his assistance.

Most especially, I want to thank my wife Chris and my daughter Jamie, to whom this work is dedicated.

D.D.C.

The University of Texas at Austin
Austin, Texas
May, 1981

Table of Contents

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

1. Introduction

1.1 General

During the 1970's significant advances were made on computational techniques for determining optimum solutions for network flow problems. It is now possible to solve problems of tens of thousands of variables using only seconds of time on large modern computers. These advances, along with their historical precursors are described in several recent books on the subject which include Minieka (1978), Kennington and Helgason (1980) and Jensen and Barnes (1980).

Along with the computational advances, network models have been applied to a wide range of problem situations. In particular, several water resource applications are reported by the Texas Department of Water Resources (Texas Water Development Board (1974a, 1975)) and Jensen et al. (1974). This report deals with the application of a new network model formulation as applied to a water resources system.

Optimal operation of a system of interconnected water reservoirs is an important problem in water resources management. The limited water resources available coupled with the diverse, often competitive, projected demands on these resources appear to place potentially unacceptable limits upon the achievement of economic, social and environmental goals.

The operation of a muliperiod multireservoir system requires that the system controller make decisions regarding the storage or release of water for each of the reservoirs on a periodic basis. This period may be daily, weekly, montbly, etc. His decisions may be based upon the amount of water available to him in each of the reservoirs, the types of demands for water from the various users and upon his anticipation of the future availability of water. Each period lends itself to a network representation similar to the one-period, three-reservoir system shown in Figure 1-1.

The amount of water available for distribution is a function of the amount of water stored from the previous period, the amount of inflow from runoff or from upstream releases within the period and any purchases from outside sources. Most work to date has treated the amount of water from runoff and other flows into or out of the system deterministically. That is, all data and parameters of the models are assumed to be known with certainty. Thus the models represent a decision problem in which the decision maker is faced with a great deal of complexity but no uncertainty. The complexity makes the decision problem difficult in itself. If there is uncertainty in the real situation it is often simply ignored by the model.

The multiperiod deterministic model assumes that the system

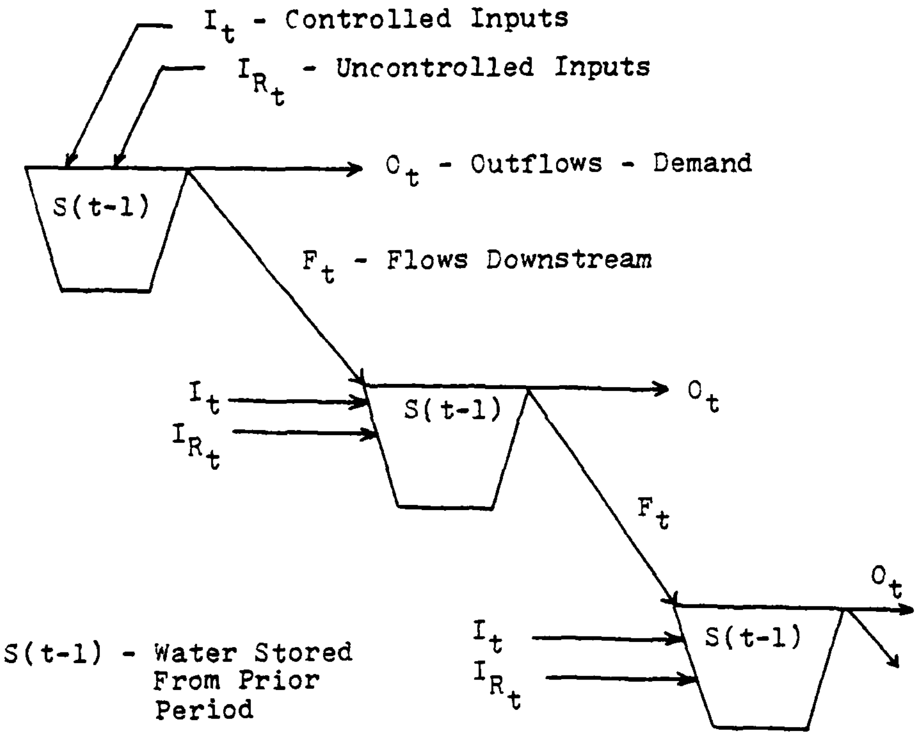

I_t - Controlled Inputs
I_{R_t} - Uncontrolled Inputs
O_t - Outflows - Demand
S(t-1)
F_t - Flows Downstream
I_t
I_{R_t}
S(t-1)
O_t
F_t
O_t
I_t
I_{R_t}
S(t-1)
S(t-1) - Water Stored From Prior Period

Figure 1-1
Three Reservoir Problem

controller knows what the future availability of water will be with certainty. He also assumes that the demands placed upon the system are known. With this information available to him he can then determine an optimal set of decisions for the entire time horizon.

The neglect of uncertainty results in unrealistic solutions where a major aspect of the decision process relates to dealing with uncertainty. The water resources problem obviously is dynamic in nature in that decisions must be made sequentially over time. Uncertainty plays a significant role in the decision process due to the unpredictabiltiy of nature in its supply of surface waters to the system and also to the incomplete predictability of the actions of man in his use of the available resource. It is clear that the controller of the system must exhibit caution in setting reservoir levels and river releases so that unlikely but possible natural events do not cause the system to fail in its functions of providing a reliable water supply and protection against floods.

A deterministic model does not exhibit caution in a dynamic, multiperiod model. Since all data is assumed certain, the future in all required detail is known. An optimum solution can be determined which provides maximum benefit at minimum cost. A historical sequence of water runoff and demand data might be used to give the model "realistic" data. The one aspect of the solution procedure which is not realistic for the multiperiod model is that the deterministic solution algorithm has the ability to look ahead in time and prepare for the events which are to occur. Thus, decisions obtained through the models do not tend to resemble the real decision process. The decision maker, after all, does not have this "look ahead" capability of the algorithm. He must make decisions in the face of uncertainty and revise them as time goes on and as uncertainties become resolved. The incorporation of uncertainty into a water resources model is a part of this research.

Another problem with most existing models is that traditional planning methodology has generally been directed toward the analysis of projects individually in an effort to match reservoir operation with anticipated requirements. When the interaction of individual reservoirs became more pronounced and could not be ignored, operating criteria were often still selected on the basis of these single-project analyses through coordinated single-reservoir simulation studies. It is noted however, that in a serially connected system of reservoirs, the value of water stored in a particular reservoir is affected by the amount of water stored in other reservoirs. This relationship has been neglected in the past due to the fact that this interaction between reservoirs suggests a nonseparable benefit function as a function of all the reservoirs versus a separate benefit function for each. This means that the total benefit of the system cannot be measured simply by summing the individual reservoir benefits as a function of their contents. One reason for this neglect lies in the difficulty in determining with any confidence just what this joint function should be. The benefit function that will be used herein is a nonseparable quadratic function which measures or reflects the current and future value of water stored by the system. Thus, besides individual reservoir benefits, the interactive or joint reservoir benefits will also be evaluated.. This idea combined with the multiperiod decision process is used in a dynamic programming approach to successively generate these benefit functions.

This report describes a method to overcome the deficiencies of the deterministic solutions while including the provision for evaluating interactive reservoirs. Network models are still used but the model is changed in such a way as to exhibit characteristics of the true decision process. Full advantage is taken of the network structure of the problem by utilizing extensively the computational techniques that have been so successful for deterministic models. Embedding this network model in a dynamic programming solution approach which begins at some specified and finite future date and works backward in time to the present provides the necessary data to allow the derivation of successive benefit functions which reflect the future value of a given configuration of reservoir contents.

Chapter 3 describes the deterministic network models and provides a brief survey of the computational techniques used to solve them. Chapter 3 also provides the notational basis for the remaining chapters of the report.

In Chapter 4, the dynamic programming algorithm for solving the multiperiod multireservoir stochastic problem will be presented. Chapter 5 includes the nonlinear network solution methodology. These network solutions embedded in a dynamic programming methodology provide the basis for deriving a functional representation of the future value of water in the face of uncertainty. In Chapter 6 the statistical aspects of the problem will be addressed.

Chapter 7 includes some example applications which are supported by data contained in the Appendix.

1.2 Primary Contributions of This Research

One contribution of this research is the development of a generalized network model which is capable of solving network type problems where some of the arcs have nonlinear quadratic cost functions. These functions are allowed to be nonseparable and the model is solved without reverting to piecewise approximations for the arc costs. The only restriction is that the overall objective function which is a combination of several linear terms along with some quadratic terms be a CONVEX cost function; or a CONCAVE benefit function in this case. It is noted that there exists several other techniques which could be used for this class of problems. Some of these are listed here:

1. Frank-Wolfe Method

2. Convex Simplex Method

3. Method of Feasible Directions
4. Gradient Projection Methods
5. Quadratic Programming Algorithm
6. Reduced Gradient Method
7. Newtons Method
8. Steepest Descent Method
9. Variable Metric Methods
   - Davidon Fletcher Powell
   - BFGS (BROYDEN,FLETCHER,GOLDFARB,SHANNO)

Most of these methods could be used either directly or in a specialized manner for this class of problems. The choice of introducing network theory as still another way to solve problems of this nature both expands and enhances the power of network theory as well as providing an alternative to the above suggessted methods.

The second and primary contribution of this research involves the integration of this network solution technique into a much larger dynamic programming model. This larger model is used to solve multireservoir multiperiod water resources problems in the presence of uncertain inflows. Its uniqueness lies not only in its use of a new network subproblem, but also in the manner in which it utilizes future stochastic runoff information and recursively generates benefit functions which represent the net current and future expected benefit to the system as a function of the observed current water levels. This functional representation allows any level to be evaluated for determining the current decisions. This form represents a continuous spectrum for current levels in lieu of the more common discretized levels in standard dynamic programming algorithms. This means that the current levels are not required to be equal to or rounded to the nearest discrete level which would induce error into the results. Although a discretization scheme is used to derive the benefit functions, it is done so only to gain a representation of the true return "functionally". Once this function is available, any and all reservoir levels can be evaluated.

One main advantage of this functional approach is that it greatly relieves the dimensionality problems associated with discrete representation of large dynamic programming problems.

In the end, the result is a realistic and usable water resources model since it does account for the uncertainties of the future. It can handle larger models due to the functional approach, and perhaps most importantly, the entire model has been implemented into a workable computer program where it is readily available to such users as the Texas Water Development Board.

CHAPTER II

2. LITERATURE REVIEW

2.1 General

Most of the research involving analytical modeling of multireservoir water systems has occured during the last 15 years. There has been a great deal of variation in the mathematical techniques employed. Roefs (1968), Buras (1972), and Hall and Dracup (1970) discuss the mathematical techniques used and the variations of the problem for which each technique is most suited. Butcher (1973) indicates that a mathematical tool useful for one water resource problem may not be suitable for other seemingly similar problems. Multireservoir models can be roughly classified into three categories depending upon their emphasis and scope.

1. Design Models. These types of models make decisions concerning the construction of the reservoir system. They are sometimes called capacity expansion models. Decisions are made concerning the size, location, and time of construction of reservoirs and canals in addition to determining water allocation.

2. Water-use Models. In these models the reservoirs are considered to be multipurpose; that is, several possible uses of water are available at each reservoir. Decisions are made

concerning such things as the timing and extent of irrigation for various crops. These models most often concern a single reservoir.

3. Time Planning Models. The main objective with these models is to determine the use and storage of water in several interconnected reservoirs in such a way as to be prepared for future shortages or excesses. This research concerns itself with this kind of model.

Typically, literature regarding water reservoir operations is characterized by four primary factors. These being:

1. System - single versus multireservoir
2. Operation - single versus multiperiod
3. Inflows - Deterministic versus stochastic
4. Return or objective function - Linear or separable nonlinear versus nonlinear nonseparable.

The mathematical models employed to model reservoir problems have included the following:

1. Linear programming
2. Dynamic programming - both deterministic and stochastic
3. Chance-constrained linear programming
4. Decomposition approaches
5. Simulation
6. Markov chains
7. Networks
8. Nonlinear programming

In most cases combinations of the above were used.

Many authors have used the above techniques to develop models for single reservoir or single period systems applying both deterministic and stochastic inflows. These works are presented briefly in section 2.2. Since this research is concerned with multireservoir multiperiod systems, a finer breakdown of models as they apply to the multireservoir multiperiod systems is discussed in sections 2.3, 2.4, and 2.5.

2.2 Single Reservoir (single and multi period) or Multireservoir Single Period Models:

Network models include the techniques used by the Texas Water Development Board (1974a, 1974b) and those mentioned below. Bhaumik (1973) presents an optimum operating policy of a water distribution system with losses (gains less than one). Concern here was with the economic effect of seepage and evaporation of water from canals and reservoirs in a water distribution sytem with reference to the Texas water plan. Weirs and Beard (1971) and Evenson and Mosely (1975) present more detailed discussions of the Texas Water Development Board's work in time planning and design models.

Linear programming has been applied to all types of reservoir problems. ReVelle and Gundelach (1975) introduced a new version of the linear decision rule in 1975. This new form permits the minimization of the sum of the variances of releases, a performance objective not previously subject to the control of the designer. The minimization of release variances of a single reservoir allows further diminishment of losses associated with deviations from target releases. They concluded that this new formulation, while experiencing definite advantages with regards to the minimum release level, had a disadvantage in that it required larger reservoir capacities. In spite of this result, this new linear decision rule makes it possible to attain release levels that otherwise might be regarded as infeasible. Gundelach and ReVelle (1975) then use this new decision rule and develop a chance-constrained model which seeks the smallest reservoir satisfying certain conditions on storage, release and freeboard.

These linear programming methods are considerably slower than the linear out-of-kilter algorithm, but they have more flexibility. For instance, monthly evaporation can be included in the model as a percentage of reservoir storage. For a multiperiod model, linear programming has the shortcommings of the necessity for perfect information, the large size of the problem, failure to make use of the final reservoir storage, wasted computation and of course the restriction of linearity. As stated by Buras (1972), "linear programming yields only point solutions in the policy space, no matter how many dimensions the space has. Most situations in which the state of the system changes (in time or in space) and in which decisions have to be taken successively are clearly outside the grasp of linear programming". A point solution means the set of optimal values for each of the variables, given fixed values for each of the parameters of the system. This differs from a functional type of solution where the solution variables are given as a function of another variable.

Dynamic programming is the most theoretically appealing approach to multiperiod reservoir models of all types since these problems involve sequential decision-making processes. Also, the outcome of each decision (or set of decisions in a time period) appears as a function rather than as a point solution. That is, the optimal decision to be made is determined for any state of the system. This allows suboptimal policies to be examined, a desirable feature due to the inherent uncertainty in multireservoir problems. This feature makes dynamic programming especially useful for real time system operation. The limitation on the usefulness of dynamic programming is the so called "curse of dimensionality". Each reservoir gives rise to a new state variable (usually the final reservoir storage level). If there are R reservoirs and each reservoir has K possible levels, there are K^R possible state combinations per time period. For this reason most of the dynamic programming models have been for systems with either one or two reservoirs.

Buras (1972) fixes four or five as the maximum number of reservoirs that can be handled computationally by dynamic programming. Dynamic programming also fails to take into account the stochastic nature of the multireservoir problems. This can be rectified by using stochastic dynamic programming as was done by

Driscoll (1974).

Buras (1972) presents a dynamic programming formulation for a design type model and for a water-use model. Young (1967) combines dynamic programming with a simulation approach.

Butcher (1973) defines stochastic dynamic programming to be those formulations of dynamic programming in which the value of one of the state variables is related in a probabilistic way to the value of that same variable in adjacent time periods. In terms of multireservoir problems, stochastic dynamic programming allows the rainfall and demand (or net demand, ie. demand minus inflows) at a reservoir in one time period to be dependent probabilistically on the net demand at that reservoir in the previous time period. The optimal policy developed is that which minimizes expected costs for the system. As with traditional dynamic programming, the optimal policy is in a form that can readily be used for real time system operation. However, the dimensionality problem is compounded due to the additional state variables which are the net demands at the various reservoirs in the previous period.

Due to the dimensionality problem, stochastic dynamic programming models have been applied mainly to water-use models, rather than other models which tend to have more than one reservoir.

Butcher (1971) presents such a model for one multipurpose reservoir. Loucks (1969) presents three stochastic dynamic programming models that he used to define operating policies for several of the Finger Lakes in New York. These are also one reservoir models. Instead of economic objectives, Loucks minimizes the sum of squares of the departures of releases from a set of target releases specified by the state.

Chance constrained linear programming is another tool that has been applied to multireservoir problems in an attempt to account for stochastic variation. Like deterministic linear programming, this method is more suitable for design or water-use models than it is for time planning models. The fact that point solutions are found, causes chance constrained linear programming to be less applicable to real-time system operation over a long time span. Loucks (1969) proposes a one-reservoir water-use model. ReVelle et al. (1975) considered the use of linear decision rules and the development of a stochastic model in 1969. In 1975, Loucks and Dorfman (1969) compared several chance constrained linear decision models for reservoir planning and operation. Their basic conclusion was that while all results of the four decision rules considered were within the constraints of the problem, all tend to yield overly conservative results. They state that linear decision rules permit the use of linear programming methods for solving what would otherwise be a very messy nonlinear stochastic optimization problem. This is indeed a mathematical advantage, but at the same time, these linear decision rules reduce considerably the number of possible operating policies that can be considered. Hence, the rule itself is a constraint imposed for mathematical reasons.

Klemes (1977) and Doran (1975) discuss the problem of selecting discrete reservoir levels and how the "curse of dimensionality" can be overcome by using a method called the divided interval technique. This technique differs from the traditional discretization method primarily in the precision with which the two boundary states are represented. The traditional method developed by Moran (1954) tends to over estimate the probabilities of emptiness and fullness, thereby underestimating the intermediate levels. Klemes and Doran show that for equivalent results, 5-10 discretizations using the divided interval technique corresponds to approximately 30 intervals using the traditional method.

The remaining mathematical methods have been less used and are not easily classified. Parikh (1966) and (1967) presents a linear decomposition method designed for use in a northern California system. Young (1967) combines dynamic programming with Monte-Carlo simulation of stochastic inflows. Su and Deininger (1971) present a Markov-chain approach for serially connected reservoirs. He solves the Markov system by a method of successive approximations rather than by dynamic programming.

2.3 Deterministic Inflows and Linear or Separable Nonlinear Objective Function:

Within this category of multiperiod multireservoir systems many techniques have been used. Several authors used a linear programming approach. Drobny (1971) was concerned with water quality and quantity problems. Salcedo (1972) dealt with a water-use model. Mannos (1955) was concerned with the efficiency of operation of a system of dams and Mejia et al. (1974) evaluated multireservoir operating rules using linear programming.

Schweig and Cole (1968) used dynamic programming to deal with random inflows having first order serial correlation. The distribution of these random inflows was approximated by discrete probability space. This correlation was simplified by classifying the inflow data according to whether an item was preceded by an inflow higher or lower than mean for the antecedent month. Rood et al. (1973) presents a dynamic programming model for the time-planning variety that is especially designed for serially linked reservoirs thereby reducing the state space dimensionality problem.

Fults and Hancock (1972) use state incremental dynamic programming to find the optimal operating policy for a four reservoir system. The objective is to maximize power generation while satisfying firm water contracts, enhancing environmental aspects and providing flood control.

Heidari et al. (1971) and Meredith (1975) use a technique called discrete differential dynamic programming (DDDP). This is an iterative method that eases the state dimensionality problem by starting with a trial trajectory satisfying a specific set of initial and final conditions and applies Bellman's recursive equation in the neighborhood of this trajectory. At the end of each iteration a logically improved trajectory is obtained and used as the trial trajectory in the next step.

Becker and Yen (1974) use a combination of linear and dynamic programming for the optimization of real time operations of a multireservoir system and Hirsch et al. (1977) combine linear programming with simulation techniques.

Prekopa et al. (1968) address serially linked reservoir design by attempting to meet all demands with a given high probability. Their objective is to minimize the sum of the building costs and penalties incurred for unsatisfied demand. Their method of solution uses a sequential constrained minimization technique (SUMT) with a logarithmic penalty function. A disadvantage of this approach is that under certain circumstances not all demands can be met with the desired probability.

Jensen et al. (1974) represent a multireservoir multiperiod system using networks. Here, the network for each period stays the same with inclusion of a set of storage arcs to join the networks from one period to the next. Kerr (1972) combines linear programming and the out-of-kilter algorithm and applies these to the Saskatchewan-Nelson river basin in Canada. He compares multireservoir analysis techniques by considering 53 possible future storage reservoirs and 22 diversion possibilities.

Hirsch, Cohon and ReVelle (1977) developed a hypothetical design for the sizing of three reservoirs in parallel. They recognized that there are benefits due to the joint operation of a system of reservoirs in excess of the benefits from optimal individual operation. Basically, they concluded that within reasonable limits any combination of three reservoirs whose capacities sum to the same total capacity has nearly the same maximum system yield. Their objective was to meet all demands, which were deterministic, while not accounting for spillage or evaporation. The method of solution involved simulation of five years of actual data and a linear programming optimization model.

Windsor and Chow (1972) present a mixed linear programming model with integer variables that is a combination design and water-use model. The linear programming method appears to be more suitable for models of these types where the number of time periods can be kept to a minimum. They consider their model a practical one computationally, but admit its weakness in not considering the stochastic nature of the problem. Meier and Beightler (1967) use a decomposition method for branching multi-stage water resource systems.

In the area of separable nonlinear objective functions, Lee and Waziruddin (1970) use two approachs, the gradient projection and conjugate gradient methods. They consider the profit accrued from irrigation and the benefit received from recreation to be quadratic functions. Roefs and Bodin (1970) use separable programming with Dantzig-Wolfe decomposition.

2.4 Deterministic Inflows, Non-Separable Objective Function

In this category, Gagnon et al. (1974) use a generalized reduced gradient approach to a very large hydroelectric system. Lasdon (1976) and TVA (1974) also use generalized reduced gradient methods as applied to water resource systems. Trott and Yeh (1973) use a stepwise state variable incremental dynamic programming approach and TVA (1974a) used a dynamic programming successive approximation approach.

Lui and Tedrow (1973) use dynamic programming and a multi variable pattern search. This multi variable polynomial objective function represents the current and future economic losses to the system. This function is determined by regression analysis where the states or reservoir levels are the independent variables and the functional return is the dependent variable. This method of representing future economic losses functionally tends to eliminate the problem of dimensionality. They use a random sampling technique to assure unbiased and efficient selection of initial state variable level combinations. In Rosenthal (1977), a multiperiod multireservoir release scheduling for maximum hydropower benefit was formulated by the Tennessee Valley Authority as an optimization model with a nonseparable nonlinear objective function and linear network constraints. Rosenthal presents a new solution technique based on reduced gradient techniques and on primal linear network flows. An unusual feature of the algorithm is an integer programming subproblem whose exact solution determines the search directions. Test problems were run on a six reservoir TVA system. His network is somewhat unique and constrained in that the system is required to be an arborescence. An arborescence is a tree with the property that no two arcs are directed away from the same node, and a tree is a connected loopless network. Thus, in this case, no provisions are made for pipeing or channeling water to other locations. All water flows downstream to the next reservoir in series.

2.5 Stochastic Inflows and Linear or Separable Nonlinear Objective Function

Consideration of the stochastic nature of inflows to reservoir systems for multireservoir multiperiod systems has just recently begun to attract attention. Sobel (1975) analyzes the structure of optimal policies for several discrete time control models of reservoir storage using dynamic programming. Most of the models considered are stochastic and are prompted by operating problems of regulating the amounts of water discharged from reservoirs. He develops an analogy between models of multiple reservoir systems and of multi-item inventory models. Pinter (1976) uses a stochastic dynamic programming method. Driscoll (1974) uses a stochastic dynamic programming approach to a multireservoir multiperiod problem that uses a revised nonlinear out-of-kilter algorithm developed by Jensen and Reeder (1974) at each stage storing the results in a benefit matrix. By assuming a cyclical pattern he repeatedly cycles through the periods until convergence of the benefit function is attained. This revised out-of-kilter algorithm allows convex functions and a limited type of nonseparable cost functions. This method allowed him to consider systems with five reservoirs without too much difficulty.

Chu (1980) developed a method to deal with stochastic situations with recourse. His work involved a two stage decision process whereby an initial decision was made based on expected demands and then as actual demands became known, a second decision (the recourse) was made to satisfy all demands. His objective was to minimize the sum of the costs from the first decision and the expected penalty costs as required by the recourse actions.

Roefs (1968) presents a stochastic dynamic programming formulation for one and two reservoir systems.

Turgeon (1980) uses two mathematical manipulation techniques to solve problems too large for dynamic programming.

1. The one at a time method, which breaks up the multi-variate problem into a series of one state variable subproblems, and
2. Aggregation/decomposition method which breaks up the N state variable problem into N subproblems of two state variables.

Both of these methods are then solved using dynamic programming.

He applies these methods to a six reservoir system where the inflows are assumed normal with a mean and variance corresponding to those of the Quebec river historical data.

Joeres et al. (1971) considered chance constrained linear programming in conjunction with linear programming techniques in deriving operating rules for joint operation of raw water sources. Curry and Helm (1972) present a chance constrained model for a single multipurpose reservoir and then Curry et al. (1972) extend this to a system of linked multipurpose reservoirs. They allow the unregulated inflows into each reservoir at each time period to be stochastic with a known probability distribution. There is independence between the reservoirs and for the same reservoirs in different time periods. They show how their formulation reduces to a deterministic linear programming model.

Helm, Curry and Hasan (1972) present a design model for a system of reservoirs. This formulation employs a mixed continuous and integer linear programming form that is solved by Benders decomposition method.

Sigvaldason (1976) uses simulation and the out-of-kilter algorithm and applies his model to the Trent River system in Ontario, Canada. He divides each reservoir into five storage zones and applies penalty coefficients for any deviations from ideal conditions as applied to these zones. Bodin and Roefs (1971) use separable programming and the Dantzig-Wolfe decomposition method.

Houck and Cohon (1978) utilized a sequentially explicitly stochastic linear programming model (SELSP) to determine a design and management policy for a two dam system. Their process was to sequentially solve policy and design linear programming models which are formed by specifying minimal portions of the nonlinear program. The major weaknesses of the model are high data requirements and computational burden. They propose a method of mitigating both deficiencies while explicitly retaining the interaction of the reservoir system. The system coordinated performance individual operation (SCORPIO) method provides the necessary information to evaluate the interaction among facilities in a multireservoir system. It is a way to utilize the available data efficiently and its use makes the use of SELSP models practical. Basically, SCORPIO solves the individual reservoir design and operating problems, and then system-wide performance characteristics are obtained by using expected values and correlations between streamflows at all of the sites.

Takeuchi and Moreau (1974) use a combination of linear programming, dynamic programming and regression analysis. The monthly operating decisions are given by solution of a piecewise linear program, the objective function for which consists of two parts. One, the immediate economic losses within the month, and two, the expected present value of future losses as a function of end of month storage levels. These expected losses are determned by imbedding the linear program in a stochastic dynamic program. Their loss functions were constructed in accordance with the following:

1. Diminishing marginal utility of water
2. Deficits involving high proportions of nominal municipal water use were considered catastrophic
3. High rates of deficit in low-flow augmentation also created serious damage.

They applied their model to a five reservoir system in the Piedmont region of North Carolina.

2.6 Summary of Literature

In summary, none of the literature reviewed addressed the full combination of multireservoir, multiperiod, stochastic inflows and nonseparable objective functions. Rosenthal (1980) made this same observation after having researched over 100 articles.

The chapters to follow present a new approach for this combined set of conditions. The methods employed primarily use networks, dynamic programming and regression analysis. The manner in which these are used encompasses the stochastic nature of inflows using a Monte Carlo approach.

CHAPTER III

3. The Deterministic Model

3.1 The Network Model

This chapter is used to describe the network flow model, introduce the notation to be used throughout the report, and review the solution approaches used to solve deterministic problems. The latter are used extensively in the algorithms which solve the stochastic problem as well. This chapter is taken from Chapter II of Jensen et al. (1980), co-authored by the author of this report.

A network flow model is simple in the sense that it requires very few kinds of structural elements and parameters to describe it. A complex model is constructed from imaginative arrangements of these simple elements. Figure 3-1 illustrates the basic structure of a network. The network consists of nodes and arcs. The nodes are represented by circles with the inscribed number used for identification. For general reference lower case letters such as i and j are used to refer to nodes, where the letters symbolize numeric identifiers. Arcs are the directed line segments going from one node to another. Consider the general arc k (k refers to a numeric identifier assigned to the arc) originating at node i and terminating at node j. Frequently the notation k(i,j) is used in cases where it is important to emphasize

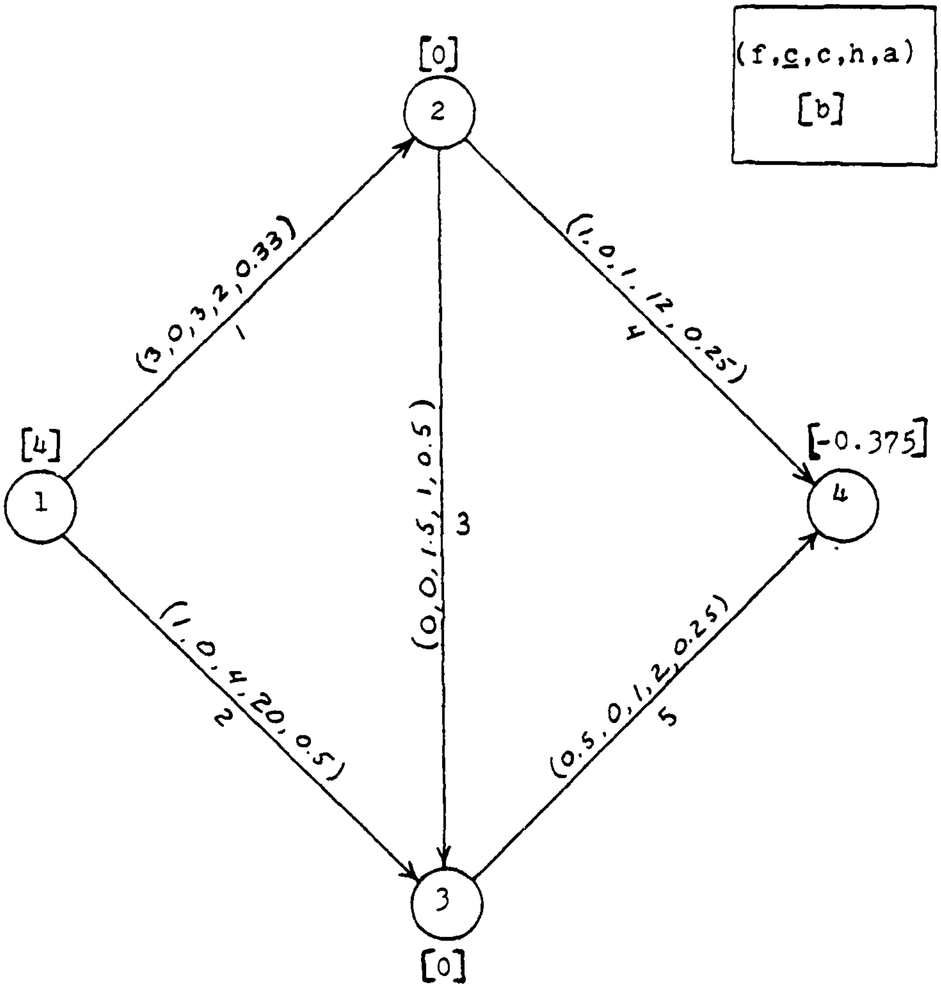


Figure 3-1
The Basic Structure of a Network

the identity of the nodes touching arc k.

The variable quantities associated with the network are arc flows. The symbol f_k represents the flow in arc k. The optimization problem is to determine the values of f_k for each arc which minimize some criterion subject to certain constraints. This criterion may be cost, time, distance, etc. The criterion and constraints are defined below.

Associated with each arc k are four parameters: lower bound on flow, $\underline{c}_k$; upper bound on flow, c_k; marginal cost, h_k; and gain a_k. Parameters and variables associated with arcs are shown in parentheses near the arc.

The values of $\underline{c}_k$ and c_k provide simple bounding constraints for the flow on arc k:

$$\underline{c}_k \leq f_k \leq c_k$$

The value of h_k indicates the marginal change in total cost with respect to f_k. In linear problems h_k is a constant independent of the value of f_k and the cost of flow on arc k is:

$$h_k f_k$$

A variation in the form of the arc cost function is presented in Chapter 5 where we will introduce a quadratic cost function.

The value of a_k, the arc gain, allows the flow to increase or decrease as it passes through the arc. For an arc k(i,j) the flow leaving node i is f_k. The flow entering node j is $a_k f_k$. Whether flow increases or decreases as it passes through the arc depends on the value of a_k. If $a_k < 1$, the flow decreases. When

$a_k>1$, the flow will increase. If $a_k<0$ the flow leaving node i with value f_k causes a flow $a_k f_k$ to leave node j in the direction of node i (on arc k). This strange possibility has some applications and is allowed by the algorithms. It is only required that $a_k \neq 0$ for all arcs. A network in which all gains are unity is called a pure network, while the presence of one or more nonunity gains results in a generalized network.

There are also parameters associated with the nodes. Node parameters and variables are shown in brackets adjacent to the nodes. The most important is node external flow, b_i for node i. This parameter represents flow entering or leaving the network from external sources at node i. Use $b_i>0$ to imply flow entering node i. If $b_i<0$, flow leaves the network with magnitude b_i . When $b_i=0$, no flow enters or leaves the network at node i. In the pure network, the sum of the flows entering the network will equal the sum of the flows leaving the network.

Additional node parameters are slack external flow, b_{si}, and slack external cost, h_{si}. If $b_{si}>0$, then flow may be obtained from external sources at node i at a cost h_{si} per unit. The amount of slack flow, f_{si} is a variable bounded by zero and b_{si}. When $b_{si}<0$, then flow may be removed at node i at a cost of h_{si} per unit. The amount of slack flow here is bounded by zero and $-b_{si}$. The sign is used only to indicate the direction of slack flow. These slack external flows are very useful modeling tools.

One additional constraint type that relates the flows in the network is the requirement that flow be conserved at each node. This means that the sum of the flows leaving a node on the arcs of the network less the sum of the flows entering the node on network arcs must equal the external flow at the node.

The linear network flow programming problem can be written in algebraic form with the definition of some additional notation as shown in Table 3-1. With the objective of minimizing costs, the problem is written as follows:

Model I

$$\text{Minimize } Z = hf + h_s f_s$$

st

$$\sum_{k \in M_{O_i}} f_k - \sum_{k \in M_{T_i}} a_k f_k - f_{si} = b_i \quad \text{for } i=1,\dots n$$

$$\underline{c}_k \leq f_k \leq c_k$$

$$0 \leq f_{si} \leq b_{si} \quad \text{for } b_{si} > 0$$

$$0 \leq f_{si} \leq |b_{si}| \quad \text{for } b_{si} < 0$$

$$f_{si} = 0 \quad \text{for} \quad b_{si} = 0$$

$$k \in M, \; i \in N$$

Table 3-1

Definition of Notation

Notation	Definition
m	Number of arcs in the network
M	Set of arcs m=(1,2,3,...,m)
n	Number of nodes

N	Set of nodes $N=(1,2,3,\ldots,n)$
M_{O_i}	Set of arcs that originate at node i
M_{T_i}	Set of arcs that terminate at node i
$\underline{c}$	Vector of arc lower bounds
c	Vector of arc capacities
h	Vector of arc marginal costs
a	Vector of arc gains
b	Vector of fixed external flows
f	Vector of arc flows
b_s	Vector of slack external flow bounds
h_s	Vector of slack external costs
f_s	Vector of slack external flows

For purposes of the algorithms to follow, two transformations which allow a somewhat simpler network model are now iescribed. In the algorithms these transformations are automatically performed by the computer programs. First eliminate slack external flows. This is done by creating a new node called the slack node at which conservation of flow is not required. The value of n is increased by one to account for the slack node, and this new node is assigned the index n. Now each slack external flow is replaced by an arc which originates or terminates at the slack node. For each positive value of b_{si} at node i, create an arc from node n to node i with capacity b_{si} and cost h_{si}. For each negative value of b_{si}, create an arc from i to n with capacity

$|-b_{si}|$ and cost h_{si}. The lower bound is set to zero and the gain is set to one for these arcs. Now the network has no slack external flow parameters but rather slack external flows are represented by arc flows to or from the slack node. The arc set is expanded to include these new arcs.

The second transformation makes all arc lower bounds equal to zero. This is illustrated in Figure 3-2. Here, for each arc k(i,j) with a nonzero lower bound, $\underline{c}_k$, adjust arc and node parameters as follows:

$$\underline{c}'_k = 0$$

$$c'_k = c_k - \underline{c}_k$$

$$b'_i = b_i - \underline{c}_k$$

$$b'_j = b_j + a_k\underline{c}_k$$

The primed parameters are the transformed parameters which will be used by the algorithm. The effect of the transformation is to make all lower bounds zero. Hence, they no longer need be considered explicitly. To recover the solution to the original problem, a reverse transformation is required after the problem is solved. Let f_k be the flow in arc k obtained by the algorithm and f'_k be the flow corresponding to the original problem. Then:

$$f'_k = f_k + \underline{c}_k$$

The cost of the solution must also be adjusted accordingly. Using the same prime notation the cost of arc k for the original problem is:

$$h'_k = h_k f'_k$$

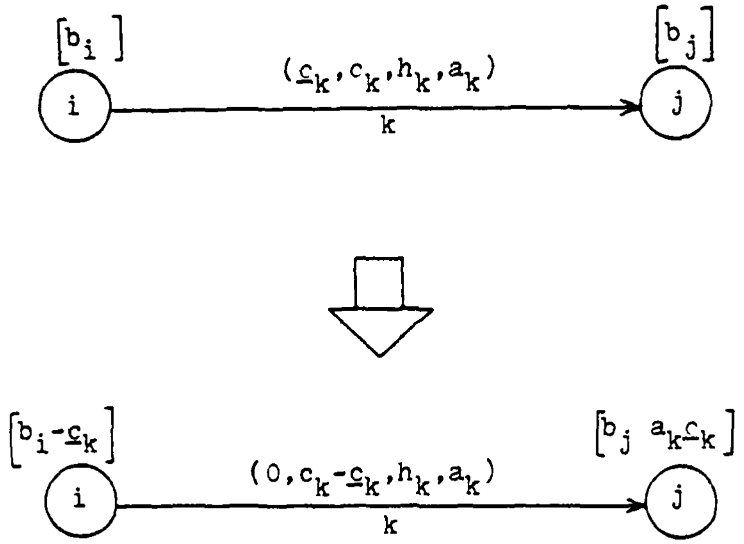

$[b_i]$
$[b_j]$
$(\underline{c}_k, c_k, h_k, a_k)$
i
k
j
$[b_i - \underline{c}_k]$
$[b_j\ a_k \underline{c}_k]$
$(0, c_k - \underline{c}_k, h_k, a_k)$
i
k
j

Figure 3-2
Transformation to Remove Arc Lower Bounds

With the transformed parameters, the minimum cost optimization problem now becomes:

Model II

Minimize hf

st.

$$\sum_{k \in M_{O_i}} f_k - \sum_{k \in M_{T_i}} a_k f_k = b_i \quad i=1,\ldots,n-1$$

$$0 \leq f_k \leq c_k$$

Note that no conservation of flow constraint is written for the slack node as this constraint would be redundant.

Model II is a bounded variable linear program. The matrix of conservation of flow constraints has only two nonzero entries for each arc, one equal to +1 and the other equal to $-a_k$. When all b_i are integer and all a_k are unity (the pure problem) the optimum flows will be integer. The flows will not in general be integer for the generalized problem.

3.2 Network Model of the Multireservoir System

3.2.1 The Multireservoir System

One of the principle ingredients which supports life, industry, agriculture and the environment in general is water. Critical to the continued preservation of life is the intelligent and efficient use of all available water supplies. This has already become a major problem for many nations of the world, and fresh water supplies are becomming more and more in demand as the population increases and as new or improved industrial techniques require it. Water sources are typically divided into ground water and surface water, and within the surface water category they can be broken down into river sources, reservoir or lake sources and perhaps even ocean sources. The models to be developed in this report deal strictly with surface water supplies in the form of rivers and reservoirs.

Many regions of the world and of the United States in particular depend upon rivers as their primary source of fresh water supplies. Areas that depend upon this form of water supply are highly dependent upon rainfall as their source. Consequently, during periods of low rainfall or drought conditions, river flow may be dangerously low, effecting both the amount and quality of water supplied. Conversely, during periods of heavy rainfall, the surrounding communities are dependent upon the river's ability to remove the excess water and to prevent serious and costly flooding conditions. For these reasons and others, dams have been constructed along existing rivers which back up the water above the dam, creating man-made reservoirs. Figure 3-3 shows a hypothetical river system with two reservoirs. The watershed for a given reservoir is that geographical area whose runoff ultimately drains into the reservoir. In the case shown the watershed for reservoir 1 includes all of the area upstream from the dam, whereas the watershed for reservoir 2 includes the area between the two dams. Reservoir 2 also receives water from the releases of reservoir 1.

Many users draw their water directly from the reservoirs rather than from the river. Since these reservoirs act as large holding tanks, the supply of water can be regulated partially through the operation of the dams. The regulation of water supply tends to reduce the possibility of low water supply conditions by storing up water for approaching dry seasons and can act as a flood control system by lowering the reservoir level prior to pending rainy seasons.

Recognizing that decisions must be made periodically (daily, weekly, monthly, etc.) as to the operation of a system of reservoirs, it is logical to consider this as a multiperiod decision problem. Within each period, decisions must be made as to how much water to supply to each of the users, how much to release downstream and thus how much to hold in the reservoir for use in the next period. The model which will be developd in this report will provide this information for the multiperiod problem.

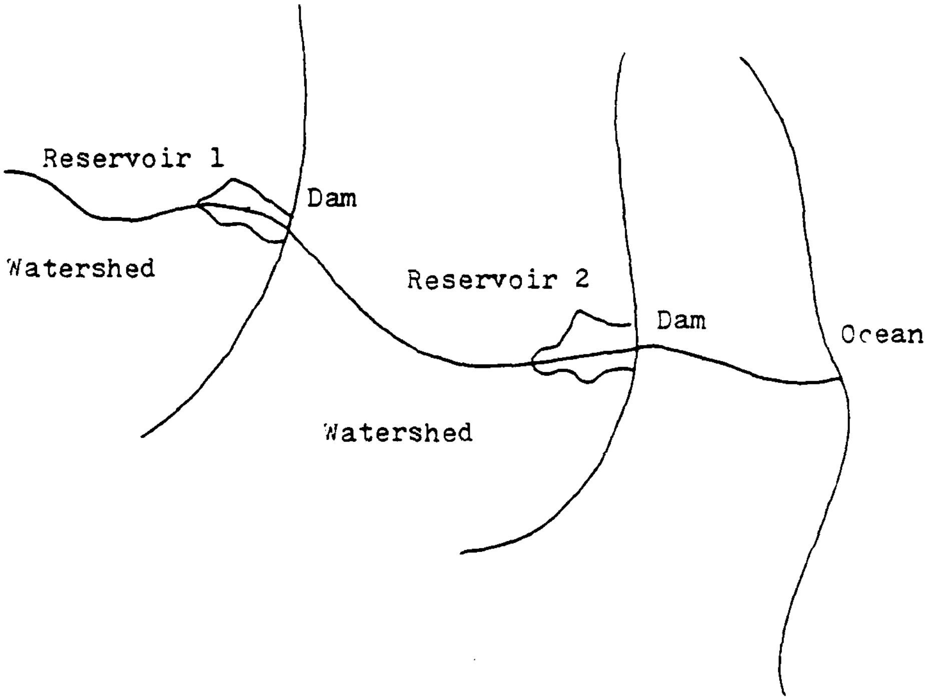


Figure 3-3
Hypothetical River with Two Reservoirs

3.2.2 The Single Period Model

In this section the network model for a single period of time is constructed. In a later section it is expanded to multiple periods. The single period model compresses all flows on a particular facility (ie., inflow, river, demand, or reservoir) into a single number which represents the total flow for the period. Thus all detail on flow variations within the period are lost. This discretizing of time is a necessary approximation to make the model of this report computationally practical. The selection of the time period is an important step in the modeling process. Different applications might lead to different selections. Thus, a time period of a day or even several hours might be necessary for the control of a flood condition, while a planning model for water supply could use a model with monthly or seasonal periods.

First the demands for water are modeled. For convenience the models of this report show all users drawing water directly from the reservoirs. All users at a particular reservoir are combined into a single equivalent user. It is easy to enrich the model for more complex arrangements by adding more nodes and arcs along the river reaches. Demand is not a fixed withdrawal of water, but rather is measured for each reservoir by a monetary benefit function for water used. Realistically one would expect that a benefit function would be a concave function exhibiting decreasing marginal return as illustrated in Figure 3-4. For this

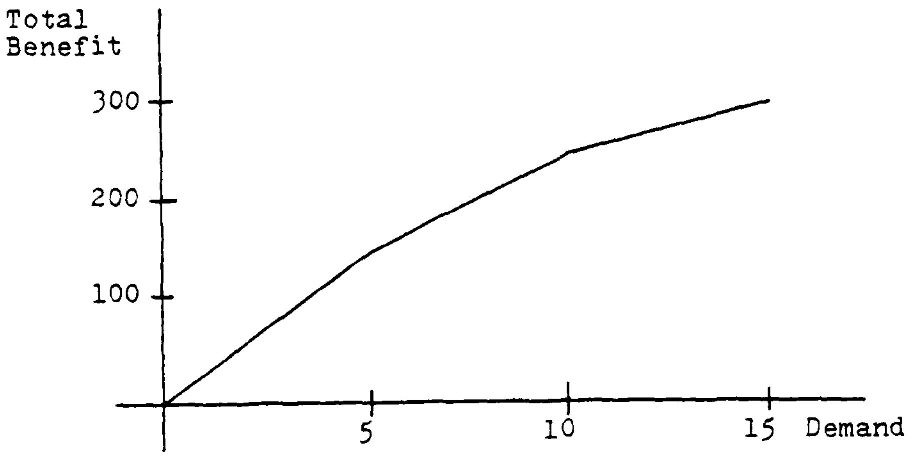


Figure 3-4
Total Benefit for Water Provided at Reservoir 1

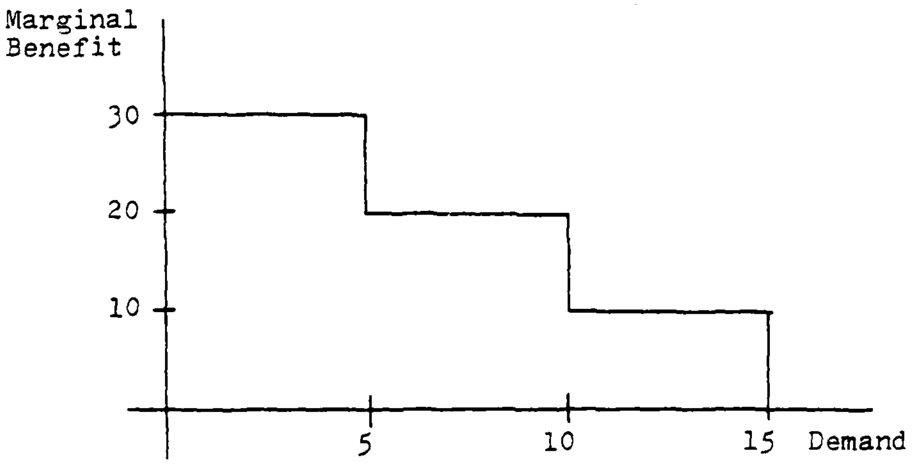


Figure 3-5
Marginal Benefit of Water Provided at Reservoir 1

report a piecewise linear approximation for benefit is used with the marginal return decreasing in steps as shown in Figure 3-5. Although it is recognized that providing a benefit function of the type described is not an easy task, it is required that one be estimated for each equivalent user.

The network model representing users for each reservoir is illustrated in Figue 3-6. Each reservoir is represented by a node in the single period model. For the example, nodes 1 and 2 represent reservoirs 1 and 2 respectively. Each user is also represented by a node. A node is also provided for the ocean. The three arcs connecting each reservoir node with the associated user node represent the piecewise linear approximation for the benefit of water to the users. Since the network model uses only cost, the benefits are shown as negative costs. Each of these arcs has a known capacity which represents the step in the piecewise function. Figure 3-5 shows the marginal benefit for water provided at reservoir 1. Thus, if 0 to 5 units are provided to user 1, the benefit is $30 per unit. The marginal benefit for 5 to 10 units is $20 and the marginal benefit for 10 to 15 units is $10. To represent the benefit function as a cost, one must take the negative of the benefit function. The cost function thus formed is convex. A negative slack external flow must be provided at each user node to allow flows to leave the network.

Another important aspect of the single period model has to do with river flow. Rivers are represented by arcs between

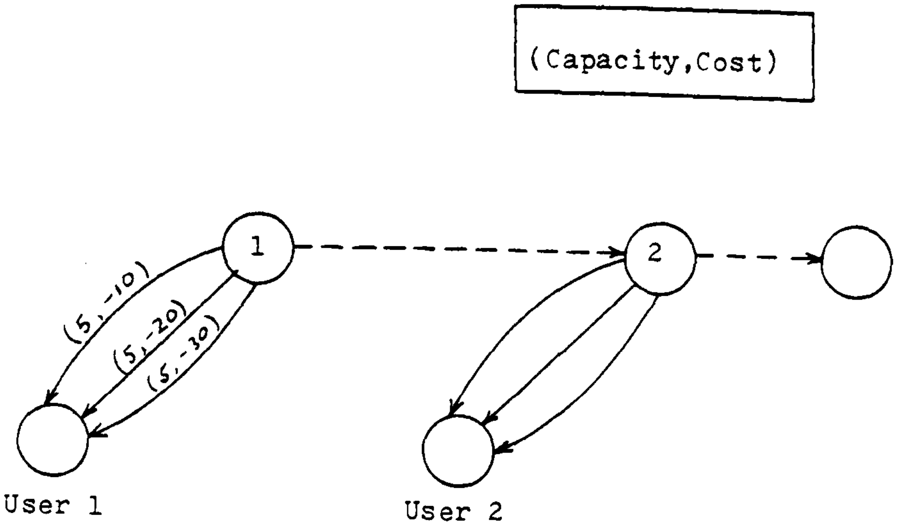


Figure 3-6
Two Reservoir System with Demands

reservoir nodes or from a reservoir node to the ocean node. Conditions may exist whereby an especially low or high flow would be undesirable. Consequently, additional arcs can be added to this network which would place a premium on meeting certain low flow conditions and a cost on high flow conditions. Network models allow a lower bound on an arc which forces the system to supply at least a specified minimum amount of flow to that arc. However, conditions may exist in which there is not enough water available to provide even this minimum amount of flow, thus resulting in an infeasible solution. Another way to handle this which is more general in nature and circumvents this drawback is shown in Figure 3-7.

Here, the arcs between reservoirs represent a piecewise linear function where each arc has a given capacity. The capacity of the arc with a cost of -10 may represent the desired minimum flow in the river during the period. The negative cost will tend to provide that flow if enough water is available in the system and if other needs (also measured by negative costs) are not more important. The capacity of the arc with zero cost would represent the safe flow levels of the river, between low flow and flood conditions. The arc with the high positive cost would represent flood conditions and its capacity should be set very large (again so as not to create an infeasible solution in the event of extremely large quantities of water available). The cost is the penalty of allowing a flooding condition. All of these arcs would

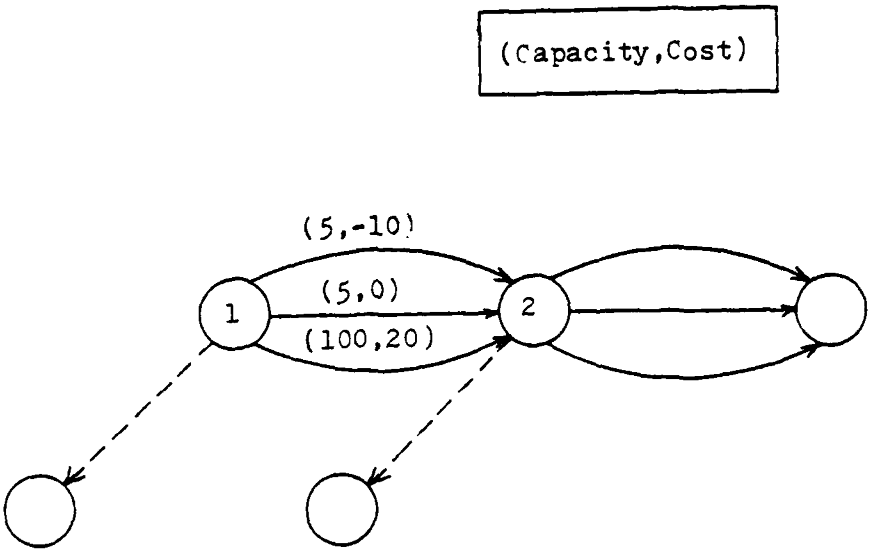


Figure 3-7
Arcs Representing River Reaches

have zero as the lower bound on flow. Naturally, additional arcs could be used which might be necessary to properly reflect the results of rapidly increasing hazardous and costly flood conditions which would obviously not be linear from the onset of a flood to a massive flood condition. These river level arcs measure costs to the current period. The total cost curve of the three arcs originating at node 1 and terminating at node 2 appears as Figure 3-8. Again, this cost function is convex.

For the single period case, the water level in the reservoirs may also be important to the area for such things as wildlife, sports and environmental issues. Of primary concern might be the quality of water if the reservoir is allowed to go too low and the safety of local areas if the reservoir is allowed to rise too high. These concerns can be reflected in the network by using parallel arcs to represent each reservoir. That is, provide arcs to indicate minimum, acceptable, and maximum reservoir levels just as was done for the river reaches. This could be done as shown in Figure 3-9. Here, the arcs from node 1 to node 1a have been added to represent (-10) the low condition, (0) the safe range and (10) the high level case. Capacities on these arcs will indicate the ranges over which the given costs are applicable. These arcs perform the very same function as the river reach arcs and their total cost curve would be the same form as Figure 3-8.

The flows in the reservoir arcs of Figure 3-9 represent the water stored at the end of the period for use in following periods.

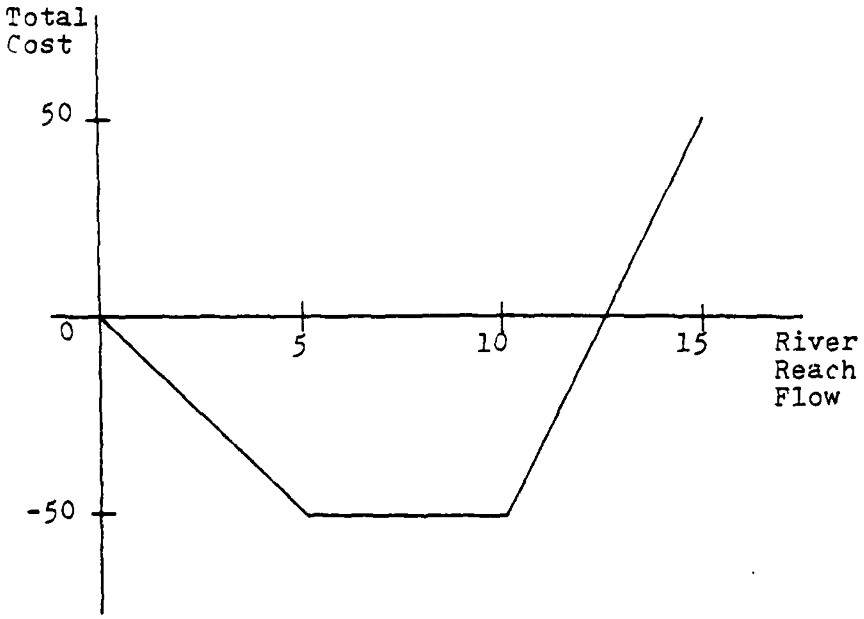


Figure 3-8
Total Cost of Flow in the River
Reach from Reservoir 1 to Reservoir 2

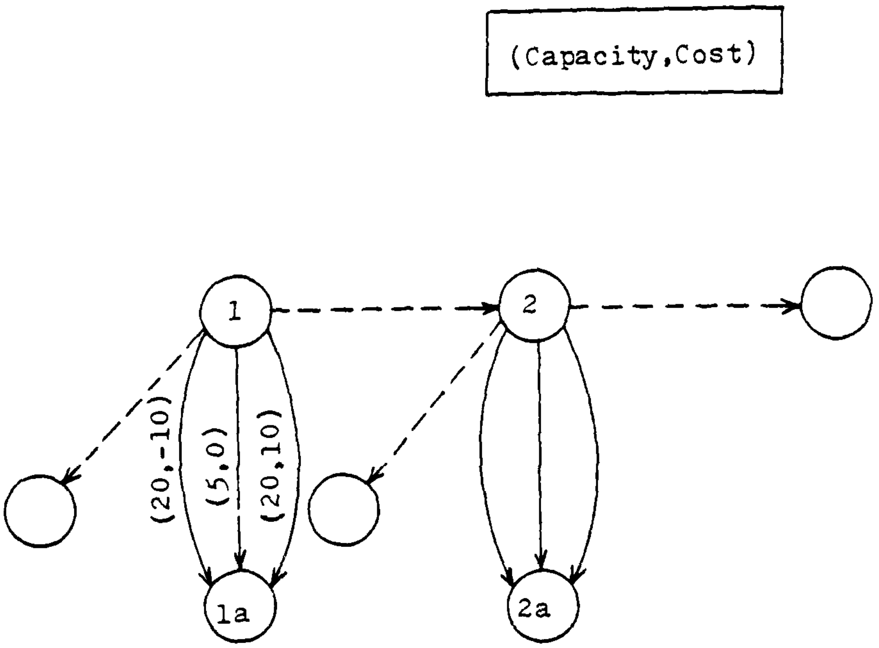


Figure 3-9
Arcs Representing Water Stored in Reservoirs

Gain factors could be used on the reservoir or river arcs to represent losses due to evaporation or seepage (the gains would be less than one). For the single period problem, water allocated to the reservoir arcs will leave the network. Thus nodes 1a and 2a would have negative slack external flows. Since the flow in reservoir arcs is limited by arc capacities, the slack external flow should be at least as large as the sum of the arcs entering the node and have a slack cost of zero. For the multiperiod case nodes 1a and 2a will be nodes in the network model of the following period.

All that remains for this single period model is to provide a source of water to the network. Inflows may be of several types, including runoff and ground seepage due to rainfall, returns from urban and industrial users and imported water as well as the reservoir waters saved from the previous period. All of these will be represented by positive fixed and slack external flows at the nodes of the network. Fixed external flows are used for inputs that are not optional and must be forced on the network such as deterministic runoff, return waters and reservoir contents at the beginning of the period. Slack external flows can be used for optional inputs such as imported water.

Note that allowing inflows at nodes discretizes the locations of the inflows. Thus although runoff and irrigation return flows are nonpoint inflows, they are approximated as point inflows. The effects of this approximation are diminished if more river reaches (hence more nodes) are defined.

The inflows to the system caused by rainfall are, of course, not known with certainty. It is this aspect of the system model that will receive the most attention in the chapters to follow. Various assumptions will be made about the knowledge of water inputs and the decision options available to the controller of the system. In this chapter, it is assumed that all external flows are deterministic and thus known with certainty. The values chosen can be the expected value derived from statistical analysis of historical runoff records or they could be specific historical sequences imposed on the system to measure the effectiveness of the system.

The entire two reservoir single period model as a network is shown in Figure 3-10. All parameters shown previously were for illustrative purposes and are not intended to represent realistic values. Future network representations will be of this form, however, in most cases the multiple arcs between nodes will be shown as a single arc for clarity.

3.2.3 The Multiperiod Model

For the single period system discussed above, it is assumed that the decision maker or system controller has access to the required data to provide the appropriate measures of benefit or cost for the various network components (demand, river levels, reservoir levels) as well as the rainfall data. It has also been

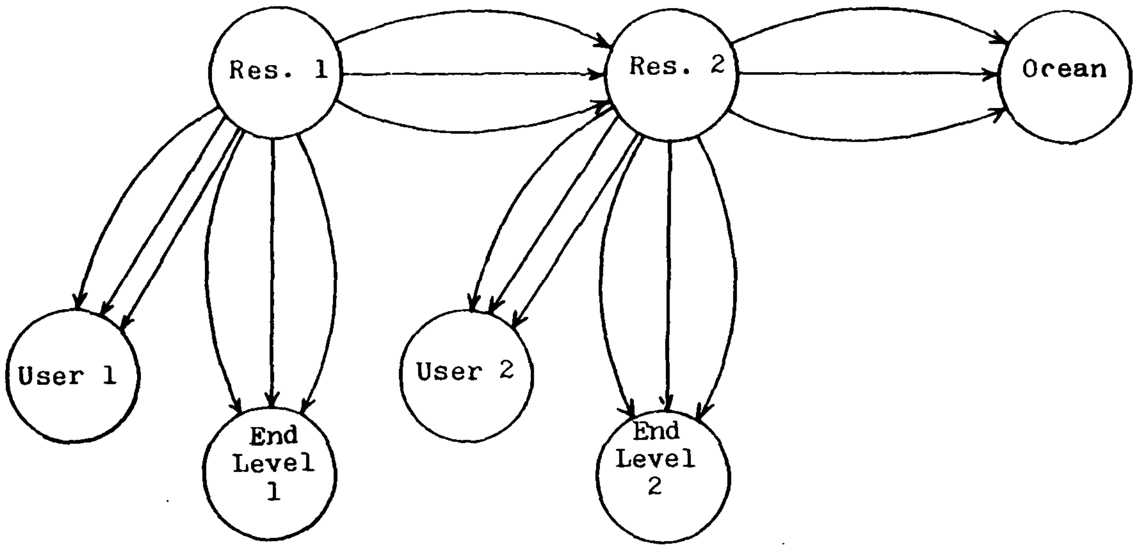


Figure 3-10
Complete Two Reservoir Single Period Model

assumed that these benefits can be represented as piecewise linear functions. These benefits are strictly in terms of current period benefits, that is, no benefits have been specified beyond the current period. For the single period problem, the decision maker observes the initial reservoir contents, calculates an expected inflow and solves the single period model to determine his optimum decisions. This process is repeated in exactly the same way for each successive period. There is a major drawback to this single period decision approach. The decisions made in one period have a direct effect on the possible options available in the next and, in fact, in several of the succeeding periods. The decisions in a given period should be made to maximize not only the current benefits, but also the future benefits. This will be done by introducing a multiperiod model that explicitly takes account of the tradeoffs between current and future uses.

The multiperiod model is constructed by providing a single period model for each period under consideration. The single period models are linked by the reservoir arcs as shown in the four period example of Figure 3-11. The reservoir arcs allow water stored at the end of one period to be used in the following periods. The amounts of water stored, represented by arc flows, are variables of the optimization. For simplicity, Figure 3-11 only shows one arc in each of the parallel arc sets of Figure 3-10.

The node-arc structure of the single period models in the combination are the same, but the external flows, arc costs and arc

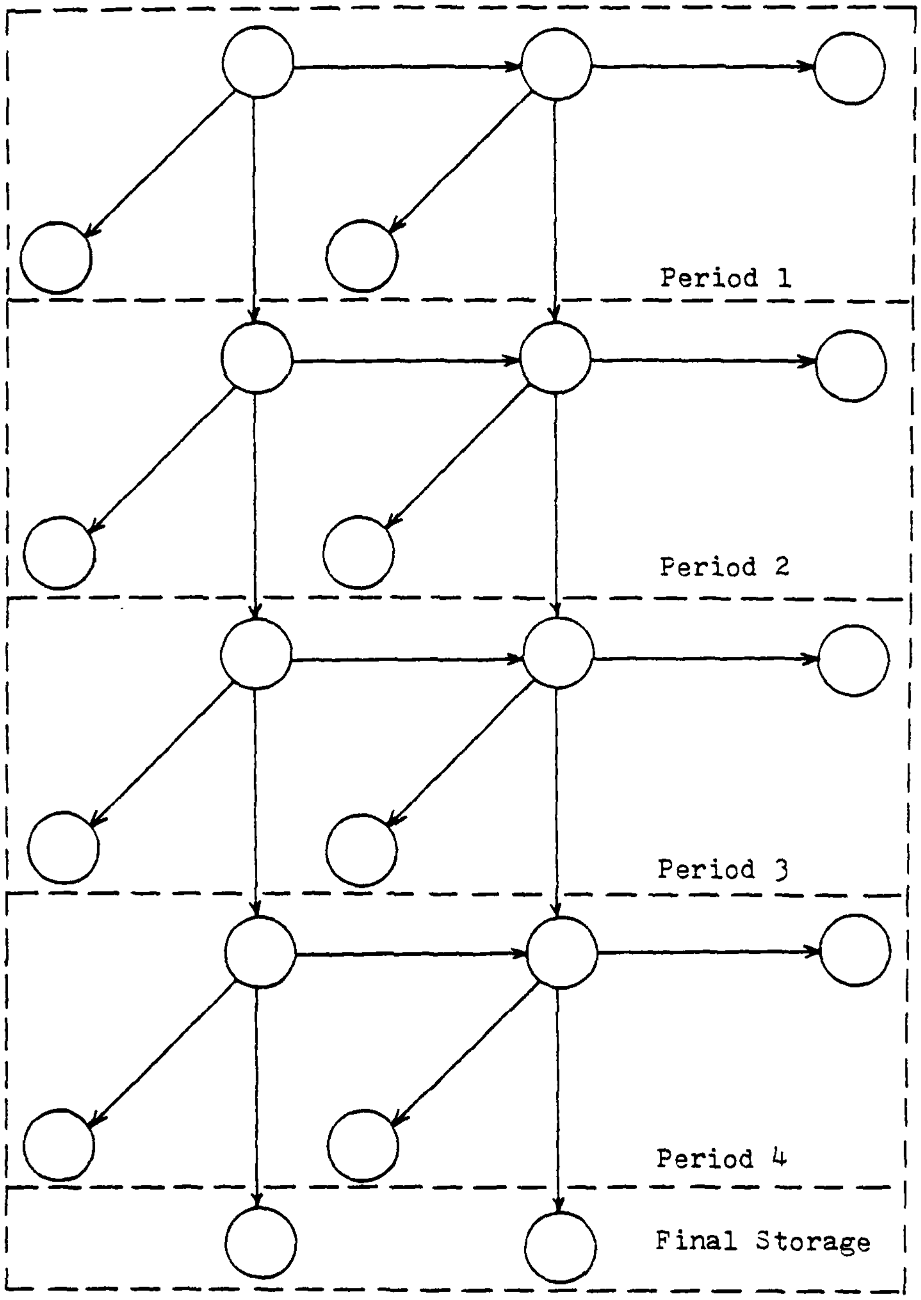


Figure 3-11
Four Period Model

capacities, which represent the supply and demand of water, will undoubtably vary over time. For instance, the example of Figure 3-11 might represent a one year time interval with the four single period models representing the four seasons. Figure 3-12 provides a more general schematic of a larger multiperiod model. Here, the single period models are represented by boxes with flows in period t given by the vector F_t. Inputs and outputs are shown by the vector quantities I_t and O_t. The reservoir arcs interconnect the periods. S_t is a vector of water quantities stored at the end of period t. Thus, S_o is the initial reservoir contents and S_T is the reservoir contents at the time horizon. For the deterministic model, all inflows must be given (perhaps in the form of a historical sequence of runoff data). Optimization of the network model will provide a policy for operation of the system in each period stated in terms of the arc flows.

One possible way of utilizing such a solution in a stochastic situation, where inflows are actually unknown, is to use the solution as a guide for policy in the first period. Then when the actual inflows are known for the first period along with the final reservoir contents, the model can be solved again to obtain a better solution for the second period. The process continues as time progresses by solving the problem for each new period as the data for the previous period becomes known.

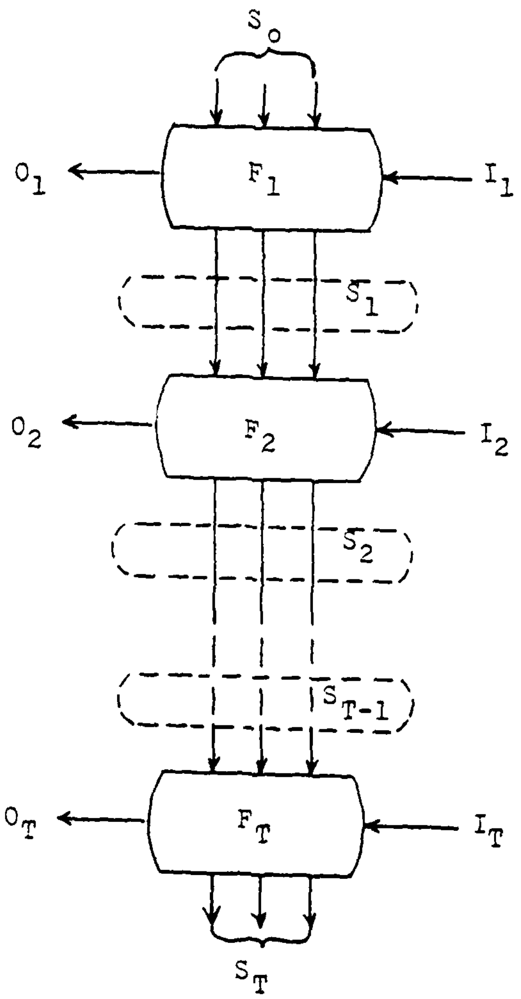


Figure 3-12
Schematic of the Multiperiod Model

3.3 Solution Procedure for the Pure Deterministic Problem

The network models which are used to describe the deterministic water resource system have the general mathematical form of Model II which is repeated here for convenience.

Model II

(1a)

Objective:

$$\text{Minimize } Z=\sum_{k=1}^{m} h_k f_k$$

(1b)

Constraints: Conservation of flow at each node:

$$\sum_{k\in M_{O_i}} f_k - \sum_{k\in M_{T_i}} a_k f_k = b_i \quad \text{for } i\in N,\ i\neq n$$

(1c)

Arc capacity:

$$0\le f_k \le c_k \quad \text{for } k\in M$$

Since this problem is a linear programming problem, the well known methods of linear programming should, and do provide a solution procedure. This section describes in general the primal simplex procedure as specialized to the network flow problem. Computer solutions of network optimization problems are described in a book Network Flow Programming by Jensen and Barnes (1980). The procedures described in the remainder of this report rest heavily on the contents of this book. This section is provided to survey the conceptual ideas of the simplex technique applied to the generalized network minimum cost flow problem.

3.3.1 The Primal Solution

Every linear program has an optimal solution which is a basic solution. For the network problem a basic solution is a selection of n-1 arcs (variables) which form an independent set. A selection forms an independent set if the columns from the conservation of flow equations (1b) associated with the set has a nonzero determinant. The basis for the generalized problem will always be a collection of a single tree rooted at the slack node and zero or more semi-trees which include a cycle. An example problem is shown in Figure 3-13. A basis for this example is illustrated in Figure 3-14. A tree is a collection of arcs on which no cycle can be formed (neglecting arc directions). A semi-tree will have a single cycle with perhaps trees rooted at nodes on the cycle. Trees and semi-trees will always be represented by directing the arcs in such a way that there is a directed path from the root to every node. This may necessitate reversing the direction of certain basic arcs. This is done by including mirror arcs in the tree. A mirror arc is given the index -k (corresponding to the forward arc k). While arc k originates and terminates at nodes i and j respectively, arc -k originates and terminates at nodes j and i respectively.

Once a basis is chosen the remaining arcs are called nonbasic arcs. A basic solution is formed by first setting the

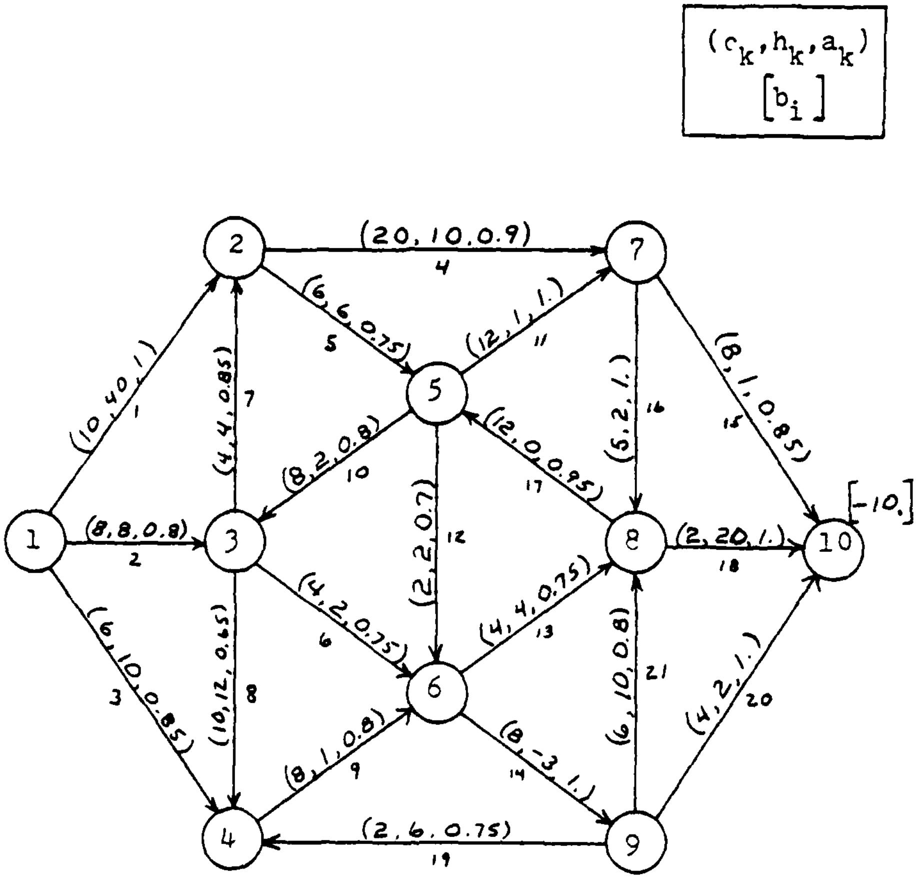


Figure 3-13
Example Generalized Network Problem

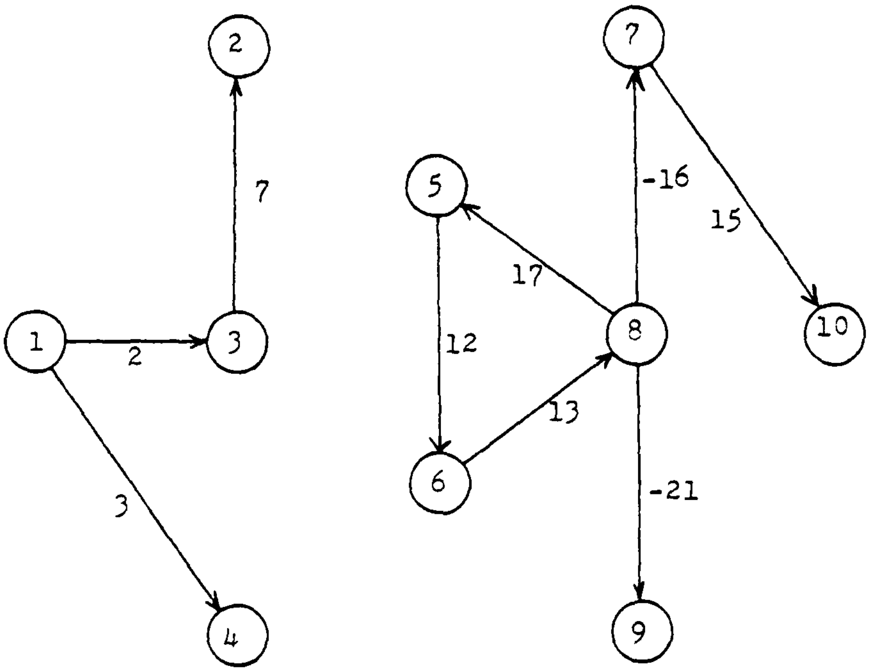


Figure 3-14
Basis for the Example Problem

flow on each nonbasic arc to either zero or to the capacity of the arc. The flows on the basic arcs are then set to values which assure conservation of flow at each node. Given flows for the nonbasic arcs, the flows for the basic arcs are uniquely determined by the external flows at the nodes. The basic solution may or may not be feasible. It is called a feasible solution if the flow on each basic arc satisfies the bounds on the arc:

$$0 \le f_k \le c_k \text{ for } k \in M_B$$

Here M_B is the set of arcs in the basis.

There are many basic solutions. For each selection of n-1 independent arcs to form a basis, there are 2^{m-n+1} possible ways to assign flows to the nonbasic arcs. There may be as many as $\binom{m}{n-1}$ ways to choose the basic arcs. Thus, an upper bound on the number of basic solutions is:

$$\binom{m}{n-1} 2^{m-n+1}$$

Linear programming theory tells us that if a feasible solution exists, at least one of this large but finite set will be an optimum solution. It is up to the optimization algorithm to find and identify which one.

3.3.2 The Dual Solution

Associated with every linear programming problem is another linear programming problem called the dual problem. The dual of the network problem will not be described here but the dual variables will be used to check a basic solution for optimality and to direct the search for an optimum basic solution. The dual variables are associated with the nodes and are called the node potentials. The node potential for node i is symbolized as π_i.

For a given basic network there is a corresponding dual solution which can be found by requiring for each basic arc k(i,j) that the following equality hold:

(2)

$$\pi_j=(\pi_i+h_k)/a_k \text{ for } k(i,j)\in M_B$$

If a mirror arc -k(j,i) is in the basis it is required that:

(3)

$$\pi_i=(\pi_j+h_{-k}/a_{-k}) \text{ for } -k\in M_B$$

Assigning the parameters to the mirror arc in relation to the forward arc k(i,j) is done as follows:

(4)

$$h_{-k}=-h_k/a_k$$

$$a_{-k}=1/a_k$$

Equation (3) combined with equations (4) yields:

$$\pi_i=(\pi_j-h_k/a_k)a_k$$

or

$$\pi_j=(\pi_i+h_k)/a_k$$

which is equivalent to equation (2).

Note that equation (2) defines a set of n-1 linear equations in n variables. Arbitrarily assigning zero as the potential of the slack node, the solution of the equations then yields the values of the dual node potentials.

Now given a basis, the primal solution (flows) and the dual solution (potentials) can be calculated. Linear programming theory provides the relations between primal and dual solutions that can be checked to ascertain optimality. These are as follows:

1. For each basic arc ($k \in M_B$) we have:

(5) Primal feasibility: $0 \leq f_k \leq c_k$

2. For each nonbasic arc ($k \notin M_B$) we have:

Complementary slackness:

(6) a. $(\pi_i + h_k)/a_k < \pi_j$ implies $f_k = c_k$

(7) b. $(\pi_i + h_k)/a_k > \pi_j$ implies $f_k = 0$

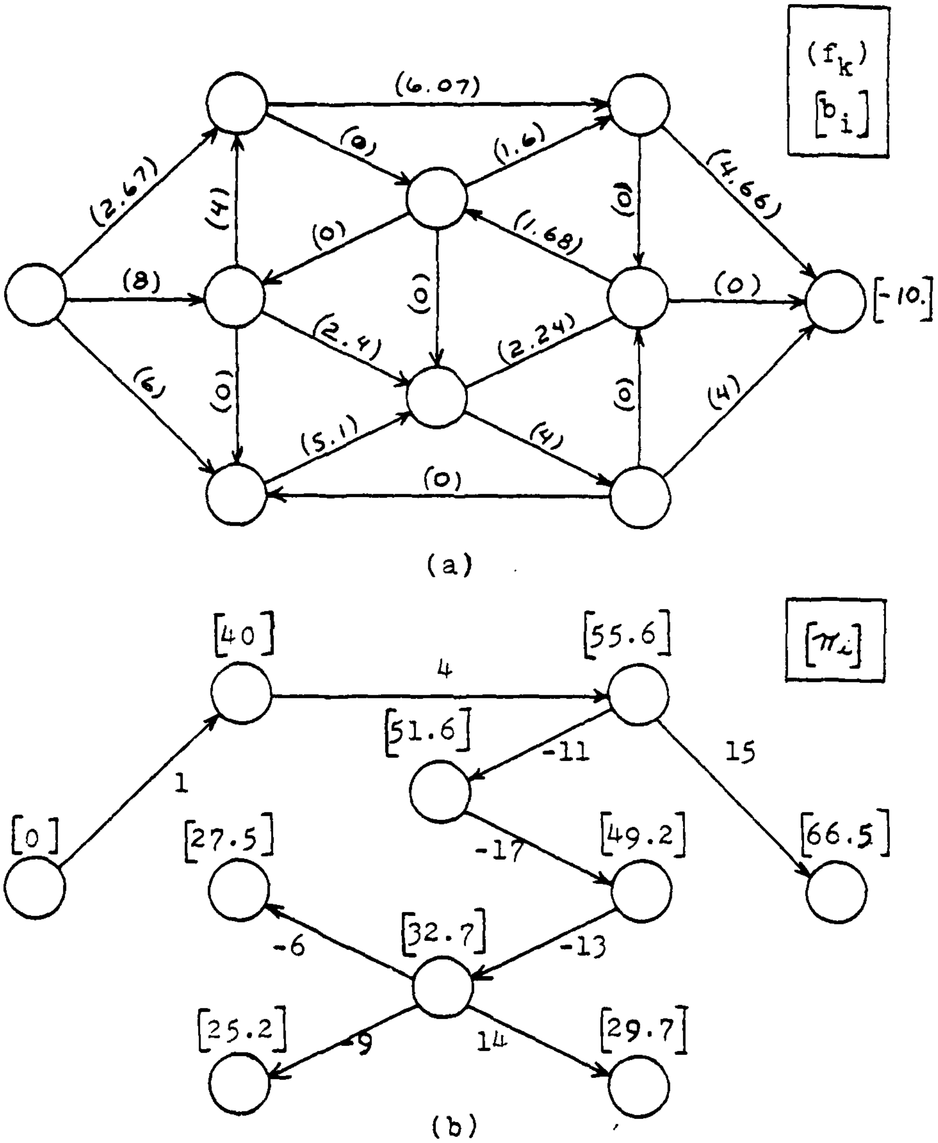
(8) c. $(\pi_i + h_k)/a_k = \pi_j$ implies $f_k = 0$ or c_k

If given basic primal and dual solutions satisfy both primal feasibility and complementary slackness both solutions are optimal for their respective problems, thus a test for optimality.

Figure 3-15a shows a flow solution for the example problem. Figure 3-15b shows the associated basic network with node potentials that satisfy equation (3). It is apparent that the flow solution is basic and feasible. It only remains to check the complementary slackness conditions. Checking these for each arc reveals that the conditions are satisfied indicating that the flow solution is optimal.

3.3.3 Primal Simplex Algorithm

Now that there is a procedure for checking optimality, a procedure is needed for directing and carrying out the search for



(a)

(b)

Figure 3-15
Optimum Solution for the Example Problem

an optimum solution. This section describes a primal solution procedure which is designed to start from a primal feasible basic solution and move in a finite number of iterations to an optimum basic solution.

First the means to start the procedure when no initial basic feasible solution is known must be provided. In general, there is no guarantee that a particular problem has a feasible solution. As frequently done in general linear programming an artifical basis is used for the network problem. Here, for each node, an additional arc is provided that connects the node to the slack node. The added arcs are called artifical arcs. These arcs are generated according to the following rules:

1. If $b_i > 0$, create an arc from i to n with capacity b_i
2. If $b_i < 0$, create an arc from n to i with capacity $-b_i$
3. For each artifical arc:
   a. Assign the arc cost of R where R is a large positive number
   b. Assign a gain of unity
   c. Assign a flow equal to capacity
4. Augment the arc set by the artifical arcs.

For the artifical solution, all flows in the original network are zero and all external flows are carried on artifical arcs to or from the slack node. Conservation of flow is satisfied for all nodes, and arc flows satisfy flow bounds. Thus, the given solution is feasible for the network augmented by the artifical arcs. The solution is very expensive, however, since all flows pass through the artifical arcs with marginal cost R. The algorithm will attempt to drive all flows off of the artifical arcs and on to the arcs of the original network to reduce the cost. If the optimal solution has nonzero flows on any of the artifical arcs it is clear that there must not be a feasible solution to the original network problem.

With an initial basic solution defined for primal and dual problems, an algorithm that can check for optimality is required and in the event of a nonoptimum solution, suggest a change that will bring the solution closer to optimality. This algorithm is the primal basic simplex algorithm which is comprised of the steps which follow:

1. Check each nonbasic arc for complementary slackness, ie.,

$$\text{if } (\pi_i+h_k)/a_k<\pi_j \text{ then } f_k=c_k$$

$$\text{if } (\pi_i+h_k)/a_k>\pi_j \text{ then } f_k=0$$

If each nonbasic arc does not violate either of these conditions, stop, the solution is optimal. Otherwise, choose an arc to enter the basis that violates one of these conditions. Let this be arc k_E.

2. For each arc in the basis, find the amount of flow change in the arc per unit of flow change in arc k_E. Use this information to find the maximum flow change in arc k_E that will cause the flow in one of the basic arcs to go to a bound or cause

the flow in arc k_E to go to its opposite bound. Choose the arc to leave the basis k_L as the arc which limits the flow change.

3. Change the flow in arc k_E and the basis arcs by the amount found in step 2. If $k_L=k_E$, return to step 1. Otherwise, change the basis tree by deleting arc k_L and inserting arc k_E. This may require some redirection of arcs to obtain directed trees and semi-trees. Change the node potentials to be consistent with the new basis network. Return to step 1.

The details of the implementation of this algorithm are fairly complex. Jensen and Barnes (1980) provide complete details. Because of the special structure of the network problem this specialized version of the simplex algorithm is much more efficient than more general linear programming algorithms applied to the network problem.

3.4 Application to the Guadalupe River Basin

Specific application of this deterministic model was made to the Guadalupe River Basin in Texas. The geographical layout of the basin is as shown in Figure 3-16.

A proposed plan for the basin is to expand the existing reservoir system to include three new reservoirs; Cloptin Crossing, Cuero I, and Cuero II. These new reservoirs along with the existing reservoir, Canyon, are felt necessary for meeting water supply demands for the future. Future demands are those projected for the year 2020 and are primarily for the San Antonio area, most

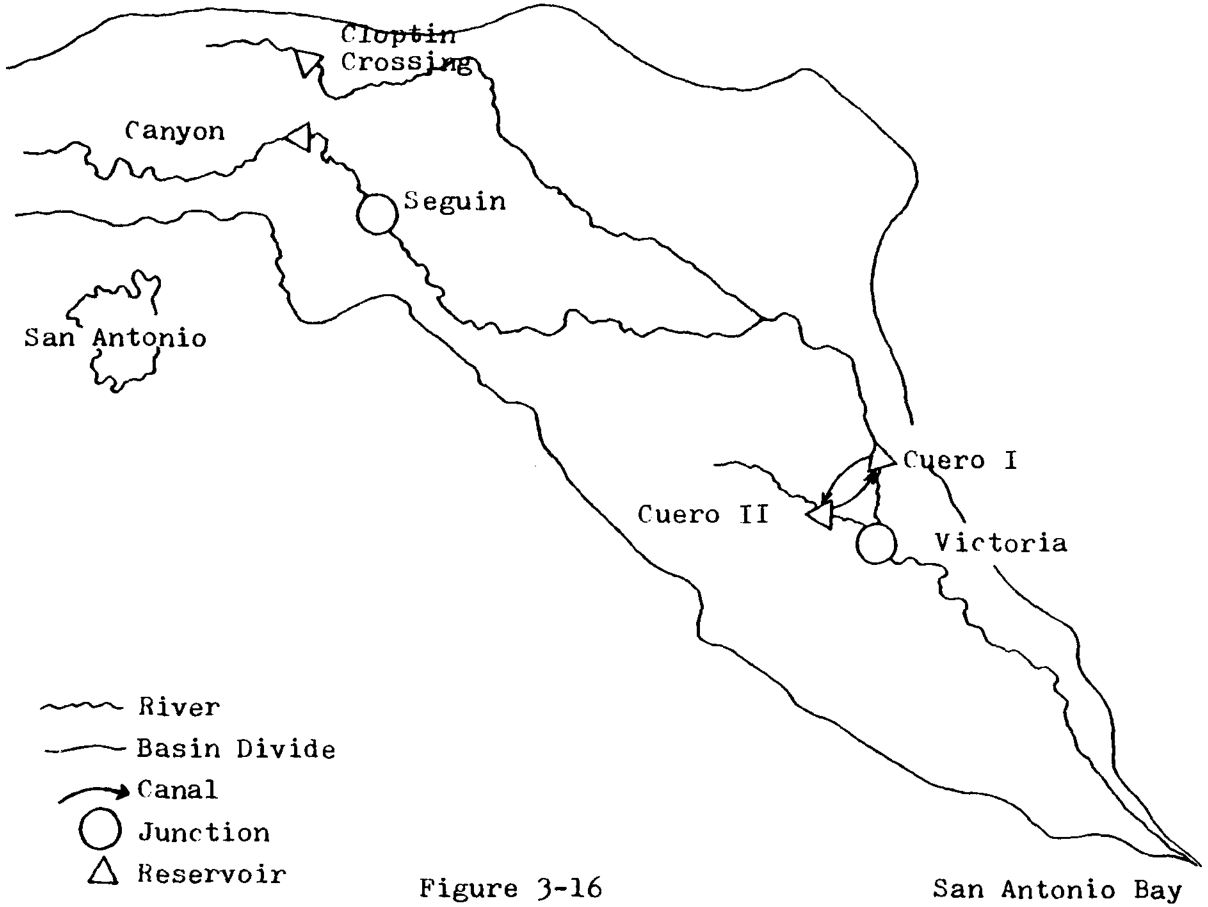


Figure 3-16
Guadalupe Basin of Texas

of which will be drawn off through Victoria.

Monthly historical rainfall and runoff data were made available by the Texas Department of Water Resources for the years 1925-1970. These data were adjusted to reflect runoff amounts into these four reservoirs, given that they had existed during this 46 year period.

This four reservoir system is shown in network schematic form in Figure 3-17. In contrast to the earlier networks, only single arcs are shown between the various nodes. This is for the sake of clarity as multiple arcs were used for this example. This network differs slightly from the previous models in that demands are allowed at junction points, Seguin and Victoria, as well as from some of the reservoirs. This was done to provide a more realistic picture of the true problem being modeled. Also, the arcs between Cuero I and Cuero II running in opposite directions imply an exchange capabiltiy due to piping and pumping where necessary. This interchange was originally planned to be an equalization channel thus implying both reservoirs would always be at the same level. If modelled this way, these two reservoirs could be treated as one.

Twelve copies of the single period model of Figure 3-17 were interconnected to form a multiperiod model with each period equivalent to a month. The multiperiod model thus represents one year of operation. The operation of a reservoir system over a period of years naturally implies that water available at the end

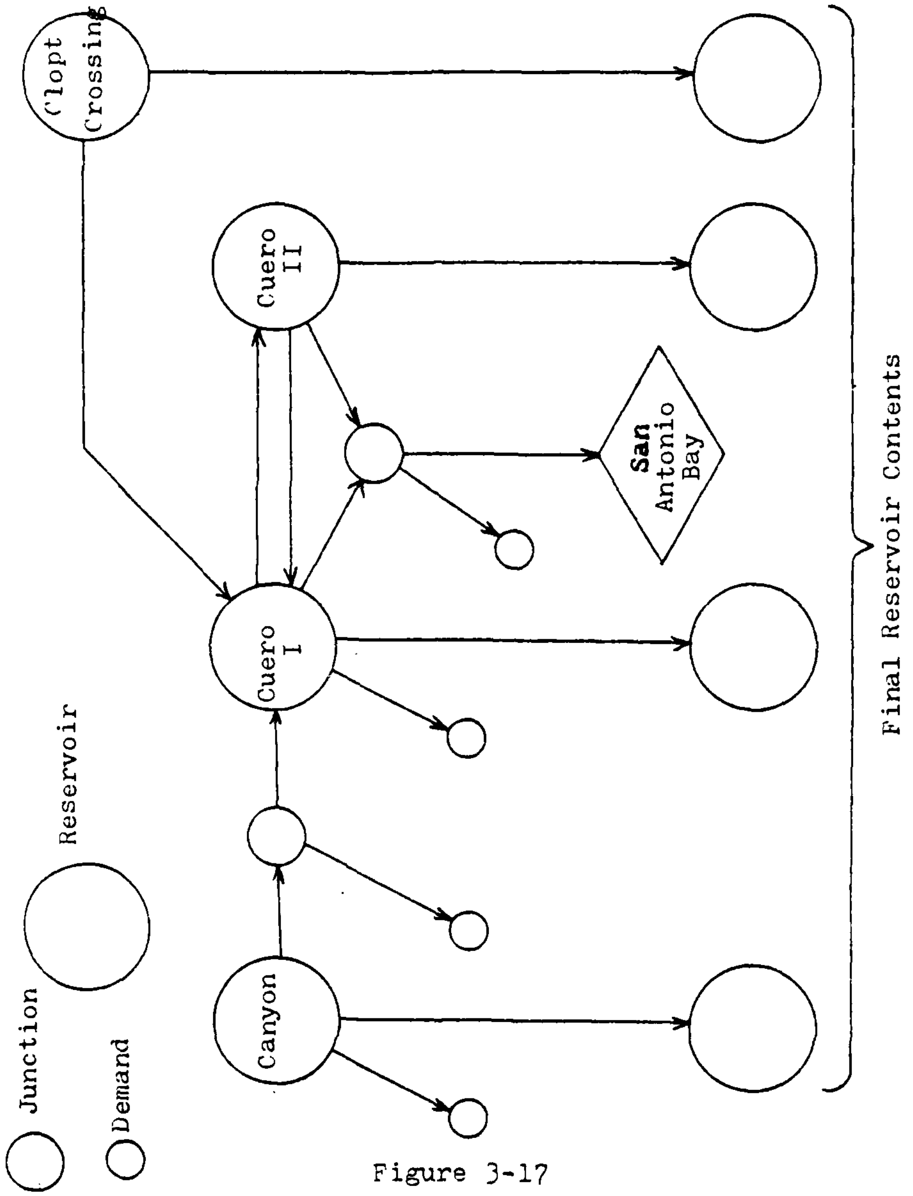


Figure 3-17
Network Schematic of the Guadalupe Basin

of this 12 month network would be available for the 13th month and so on. This is modeled by connecting the ending level reservoir nodes of period 12 with the beginning reservoir nodes of period 1, resulting in a closed or looped system of networks.

By summing the monthly inflow data and dividing by 46, the average monthly inflows were derived. These were used as the deterministic inflows to the system. Also specified were reservoir capacities, reservoir levels and demands. All data was converted to 1000 acre feet equivalents. For this exercise, demands were distributed evenly throughout the year. This most likely would not be the true state of nature for most demand points and could easily be changed to reflect a more realistic demand profile.

The deterministic solution showed that under these conditions, enough water was available on the average to meet all demands for all periods. Since there was no penalty assessed for releasing water to San Antonio Bay, and no reward for building up the levels of the reservoirs, the reservoirs tended to be held at their minimum levels. Given no penalty or reward for doing otherwise, this is what one would expect with known fixed inflows and demands. Specific node, arc and inflow parameters for this twelve period model along with the network flow results are shown in the Appendix as the Guadalupe River Basin, Deterministic Case. A matrix generator was developed to take the single period information and convert it to a multiperiod data set. At this point, the new data set could be modified to reflect changing arc parameter choices.

In the next chapter a new model is developed which takes into account the stochastic nature of runoff and the interaction between reservoirs. Then in Chapter 7 this new model will be applied to a hypothetical three reservoir system and finally to the Guadalupe River Basis four reservoir system.

CHAPTER IV

4. Dynamic Programming Solution Approach

4.1 The Decision Process

This chapter will be used to develop and present the dynamic programming solution approach to solve the multireservoir, multiperiod stochastic problem.

Consideration of a multiperiod model implies that actions taken in one period effect not only the current period but following ones as well. Decisions in the current period must take account of the impact that these decisions will have on the periods to come. To utilize this concept, a benefit function will be derived for each period in the time horizon which measures the value of water to be stored in the system for future use. Once these benefit functions are known, the network can be optimized for a given starting position with the objective of maximizing the current and future benefits.

Realistically, the benefit function for any period t should reflect the characteristic that as more and more water is made available, there is a decreasing marginal return or benefit to society. This implies that the benefit function must be concave. The model to be developed requires that this be the case.

Additionally, for a multireservoir system, one would expect some interaction to occur between reservoirs, particularly those in close proximity to each other. Thus, for systems of reservoirs, it is implied that the future benefit of water stored in a single reservoir should be a function of the amount of water stored in neighboring reservoirs.

A dynamic programming approach is used to derive an expected benefit function for water stored in each period of the time horizon. These benefits are represented as a function of reservoir contents. They are generally concave, nonlinear and may include cross-product terms that represent the interaction between reservoirs.

Because a functional approach is used instead of a discrete table, as is commonly done, the dimensionality problem usually associated with dynamic programming is partially overcome.

A network flow programming algorithm is used to solve subproblems generated by dynamic programming. Since the network problems are partially composed of nonlinear objective functions, it has been necessary to modify the network algorithms to handle this case. Details of how this is done will be presented in Chapter 5.

The first part of this chapter provides a review of deterministic and stochastic dynamic programming. This is followed by a brief description of the multireservoir network model to be used in the dynamic programming approach to the multireservoir multiperiod problem. Finally, the overall algorithm for the dynamic programming derivation of the desired benefit functions is presented.

4.2 Deterministic Dynamic Programming

Dynamic programming is a method of solving an intricate problem by decomposing it into a series of stages (of time, space, etc.) and approaching the solution stepwise. This method, also known as Recursive Optimization, is based on Bellman's "principle of optimality" (Bellman and Dreyfus (1962)) which states that an optimal set of sequential decisions has the property that whatever the first decision is, the remaining decisions must be optimal with respect to the outcome of the first decision.

In contrast to linear programming, there does not exist a standard mathematical formulation of "the" dynamic programming problem. Rather, dynamic programming is a general type of approach to problem solving, and the particular equations used must be developed to fit each individual situation. Of importance to any dynamic programming problem is the identification of the stages, the state variables, the transition equations, the decision set and the return function.

To illustrate the deterministic dynamic programming approach consider a multiperiod single reservoir problem. Here, the stages would represent the several time periods within the time horizon. A finite time horizon of T periods is assumed. Each stage would then represent a period of time for which a decision or set of decisions must be made.

The decisions to be made at each time period may be how much water to give up in the form of supplying demands and releases, or how much water to store in the reservoir for the next and future time periods.

To make these decisions, one must know the level of water stored in the reservoir at the beginning of the time period. The level of water in the reservoir is then the state variable. The ending level of water in a reservoir will be equal to its initial level plus inflows minus outflows. Let S_t equal the value of the state variable at stage t, i.e. the reservoir level at the end of period t. The value of the state variable at stage t is defined as a function of the value at stage t-1:

$$S_t = S_{t-1} + i_t - d_t$$

where:

S_{t-1} is the level at the beginning of period t

S_t is the level at the end of period t

i_t is the inflow in period t (assumed known for the deterministic problem)

d_t is the decision on outflows for period t

This is called the transition equation and it is defined for all t from 1 through T. Its purpose is to uniquely define the value of the state variable at the input to the next stage. Assume for now that the above defined terms can only take on the integer values 1,2,...,9, and that any combination of them that forces S to be less than 1 will take on the value of 1 or greater than 9, the value 9. These are referred to as boundary conditions. Hence, for each stage, the state variable (water level) can take on any of 9 values.

Having defined the stages and state variables consider now the possible decisions. Let the decision to be made at each stage equal the total outflow. Remember this may represent both water supplied to users and water released downstream. The decision can range from $\min(S_{t-1}+i_t-9)$ to $(S_{t-1}+i_t-1)$ since only available water can be released and there must be at least one unit left. For each level of S, there are at most 9 possible transitions that can be made depending on the value of i and on the decision d. Thus, between any two consecutive stages there are at most 81 possible paths to take.

Next, the return function must be defined. This represents the immediate cost of making a given decision starting from a given state. The immediate cost of making decision d_t while in state S_{t-1} is expressed as $C(S_{t-1},d_t)$. Based on Bellman's principle of optimality, the total cost function is the sum of the immediate cost plus the cost associated with making optimal decisions from the new state to the end of the time horizon. This is expressed as:

$$C(S_{t-1},d_t)+f(S_t)$$

where $f(S_t)$ is the cost of the optimum policy for periods t+1 through T. Taking the minimum of this over the possible decisions yields the desired recursive relationship:

$$f(S_{t-1})=\underset{d_t}{\mathrm{Min}}(C(S_{t-1},d_t)+f(S_t)) \text{ for } t=1,\ldots T$$

The state variable S_1 represents the reservoir level at the end of period 1, S_2 at the end of period 2, S_3 at the end of period 3, and so on. The state variable S_T is the reservoir level at the end of the time horizon. The quantity S_0 is the initial reservoir level (at time 0).

If a value of $f(S_T)$ for every possible state S_T is assumed, the recursive equation for t=T can be solved. Note that solving this equation for the example requires a minimization over as many as nine decisions for each state S_{T-1}. There are nine different values of $f(S_{T-1})$, one for each possible value of the state variable S_{T-1}.

With $f(S_{T-1})$ known, the recursive equation can be solved for t=T-1. The procedure continues until the recursive equation is solved for t=1. At this point, the solution is complete except for the recovery of the optimum solution.

The process of recovering the optimum is called the traceback procedure. Let $d_t*(S_{t-1})$ be the optimum decision found for state S_{t-1} by solving the recursive equation. Given an initial value of reservoir contents S_0, the optimum decision for period 1 is $d_1*(S_0)$. The transition equation can be used to find the optimum value of S_1:

$$S_1=S_0-d_1*(S_0)+i_1$$

The value of S_1 determines the optimum decision $d_2*(S_1)$ which in turn indicates the value of S_2. This traceback procedure ultimately leads to the complete optimum solution for the deterministic problem.

4.3 Stochastic Dynamic Programming

The above example is illustrative of a deterministic problem in that the state at the next stage is determined by the inflows during the period which are assumed known. In a realistic situation, the inflows are, of course, not known with certainty because they depend on the variability of nature. With stochastic dynamic programming, it is not necessary to assume a deterministic transition. Rather, a probability distribution on the transition is defined. Thus, let $p(S_t | d_t, S_{t-1})$ be the probability of transition to state S_t given that decision d_t is made starting in state S_{t-1}. Because some transition must be made, it is required that:

$$\sum_{\text{all } S_t} p(S_t | d_t, S_{t-1}) = 1$$

Define $C(S_{t-1}, d_t, S_t)$ to be the cost of starting in state S_{t-1}, making decision d_t and ending in state S_t. Now the stochastic dynamic programming recursive equation can be written. Let $f(S_t)$ be the expected value of starting at state S_t (at the end of period t), traversing to the time horizon, and always making the decison which minimizes the expected cost. Then:

$$f(S_{t-1})=\underset{d_t}{\text{Min}} \sum_{\text{all } S_t} p(S_t \mid d_t, S_{t-1})(C(S_{t-1}, d_t, S_t)+f(S_t))$$

for t=1,2,...,T

This recursive equation is again solved backwards by first assuming a value for $f(S_T)$ and then solving for $f(S_{T-1})$. This allows the solution for $f(S_{T-2})$ to be obtained. The process continues until the value of $f(S_0)$ is determined. As this equation is solved for each discrete value of S_{t-1}, an optimum decision is found $d_t*(S_{t-1})$. This is the decision that minimizes the expected cost from period t to the time horizon given that the system is in state S_{t-1} at time t-1.

Although the optimum decisions are known for every state value, the optimum set of decisions for the entire time horizon cannot be determined. The traceback operation previously defined for deterministic problems is not applicable for the stochastic case. The traceback is not applicable since the transition is not certain at any stage, rather, it is governed by a probability distribution. Although an optimum decision is determined for each state, only the first decision is determined since only S_0 is known. Stochastic dynamic programming models a realistic decision process in which decisions are only made for the current period on the basis of the current state value.

It should be stressed that stochastic dynamic programming uses the criterion of expected value of costs. There are other criteria which might be more appropriate such as stochastic dominance (Barnes et al. (1973)). It is not clear however how these could be incorporated into an optimization algorithm such as dynamic programming. All the literature surveyed by the author involving stochastic dynamic programming utilized the expected value criterion, so this criterion will be used in this report.

This standard approach to dynamic programming has a major drawback: the "curse of dimensionality". As the number of state variables increases, the size of the problem in terms of both computer storage and computation time becomes prohibitive. In the example above, there was only one state variable with nine possible decisions at each stage. If another state variable were added (e.g. the level of a second reservoir), there would now be 81 unique states for each stage. The number of possible states at each stage is equal to the number of levels raised to the (number of reservoirs) power. In a five reservoir problem there would be 59,049 states. Problems in water resources frequently involve systems of four or more reservoirs.

In the sections to follow, a technique is developed which replaces the discrete vector $f(S_t)$ by a single mathematical function (a benefit function) for each period, and adopts a method of sampling from the distribution of inflows. Both of these approaches greatly contribute to a reduction in the computational requirements of dynamic programming and thus allow somewhat larger systems to be solved.

4.4 Network Model for Multireservoir Multiperiod Problem

For illustrative purposes and for reference throughout this report consider a three reservoir network as shown in Figure 4-1. This model is very similar to the two reservoir model of Figure 3-10 with one exception. Here, three new arcs (1,2,3) and three new nodes (4,8,12) have been added. These new arcs will be used to represent the future value of water to the system. Their cost functions will include the nonlinear portion of the objective function and in most cases they will be nonseparable. It is the combined cost function of these three nonlinear arcs that represents the future value of water to the system since the flow in these arcs represents water stored for future use. The cost functions for these arcs will also include linear terms.

Because of the dynamic programming approach to be used, it is not necessary to connect the ending reservoir levels to the beginning reservoir levels of the next period as was done in the deterministic multiperiod case. Thus, all nonlinear arcs representing final storage could be terminated at a single node. For modeling and visual convenience, three nodes will be used rather than a single node.

The network model of Figure 4-1 is all that is required for the multiperiod model. This greatly simplifies the data input requirements. Because of the dynamic programming approach this seemingly simple single period model provides all the required

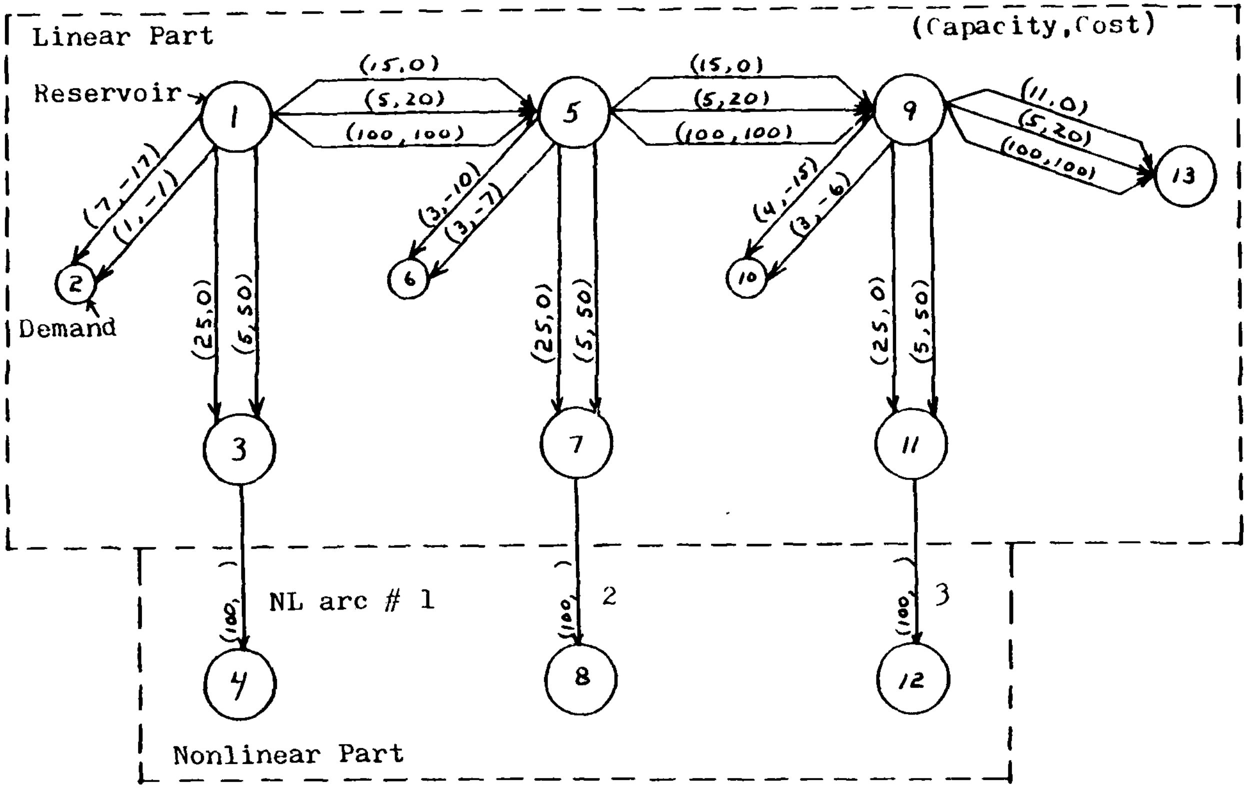


Figure 4-1
Three Reservoir Nonlinear Network

AD-A106 769 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH F/G 13/2
A MODEL FOR SOLVING MULTIPERIOD MULTIRESERVOIR WATER RESOURCES --ETC(U)
MAY 81 D D COCHARD
UNCLASSIFIED AFIT-CI-81-48D NL

2 of 3
AD
A106769

information for multiperiod use. This will become more obvious in the next few sections.

Naturally, as the model represents different periods, any or all of the network parameters can change to reflect changing water availability and requirements over time. As will be seen, even if it is not desired to change the network parameters in the linear part of the network, the cost parameters of the nonlinear arcs will necessarily change from period to period due to the changing inflow parameters causing flow changes in these arcs. A change of flow in any of the nonlinear arcs can cause a change in the cost assigned to other nonlinear arcs if the cross product terms are nonzero.

4.5 The Dynamic Programming Algorithm for Deriving the Benefit Functions for the Multiperiod Multireservoir Problem

Figure 4-2 represents the multiperiod model for the stochastic problem. Each box is a single period model of the type shown in Figure 4-1. The notation in this figure is as follows:

T = number of periods of the analysis or time horizon

R = number of reservoirs, $r=1,\ldots,R$

I_t = $(i_{1t},i_{2t},\ldots,i_{rt})$. This is a vector of runoff inflows to the reservoirs during period t. Assume that I_t is a random vector from a known distribution. The parameters of the distribution or the distribution itself may differ from period to period. The inflows for two different periods are assumed to be independent random variables. It is assumed that these are the only external flows into the system except the initial reservoir contents of period 1.

S_t = $(s_{1t},s_{2t},\ldots,s_{rt})$. This is a vector of reservoir contents at the end of period t. Note that it also describes the initial reservoir contents of period t+1.

O_t = $(o_{1t},o_{2t},\ldots,o_{rt})$. This is a vector of outflows during period t.

F_t = $(f_{1t},f_{2t},\ldots,f_{mt})$. This is a vector of arc flows in the single period network for time t. The vectors S_t

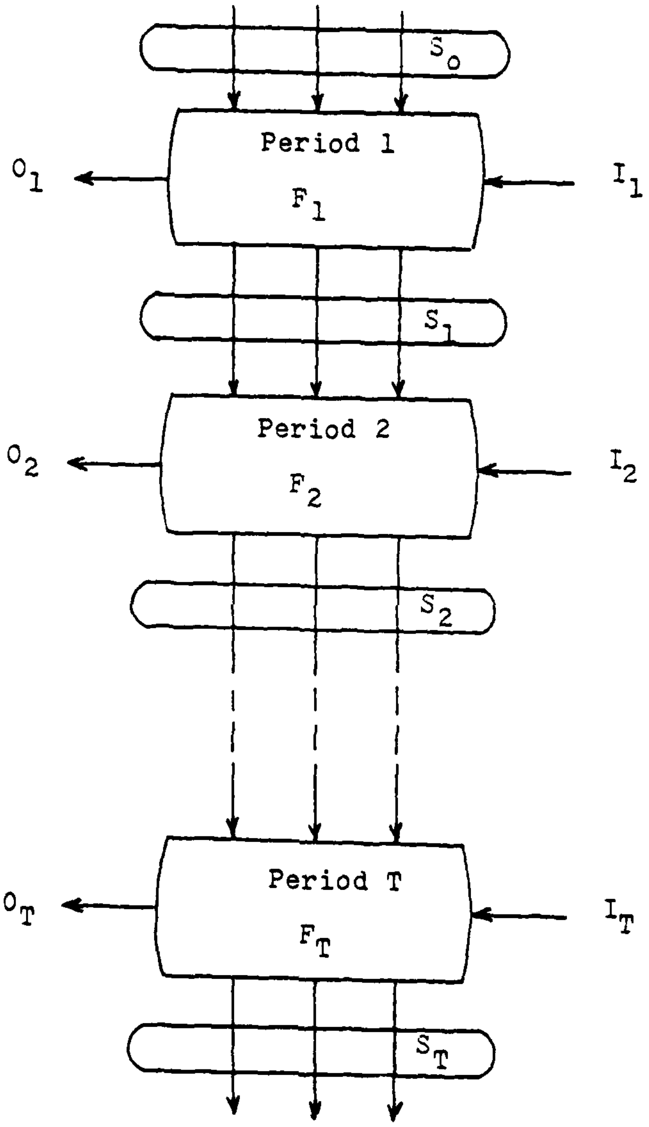


Figure 4-2
Multiperiod Water Distribution Model

and O_t are also included as flows on arcs.

The inflows to the network in period t are:

$$S_{t-1}+I_t$$

They are represented as positive external flows at the reservoir nodes. The outflows of the system are flows to the demand nodes and the end of the period reservoir contents nodes. For the single period model of Figure 4-1 it is assumed that inflows will be known before the flow decisions are made. In reality, flows are continually adjusted throughout a period as the inflows are revealed by the passage of time. For this discrete model the length of the time period may be adjusted to allow any desired degree of accuracy however the distribution of flows within a time period are assumed to be instantaneous.

The multiperiod model is solved with stochastic dynamic programming. The flow chart of Figure 4-3 represents the overall dynamic programming process used for deriving the benefit functions for all t. The periods referred to in this flow chart represent the T periods of Figure 4-2.

Before describing the algorithm in detail some additional notation is necessary.

Let:

NN = Number of discretizations of reservoir contents

CK_r = Capacity of reservoir r

RL = Vector of reservoir levels for discretization as a

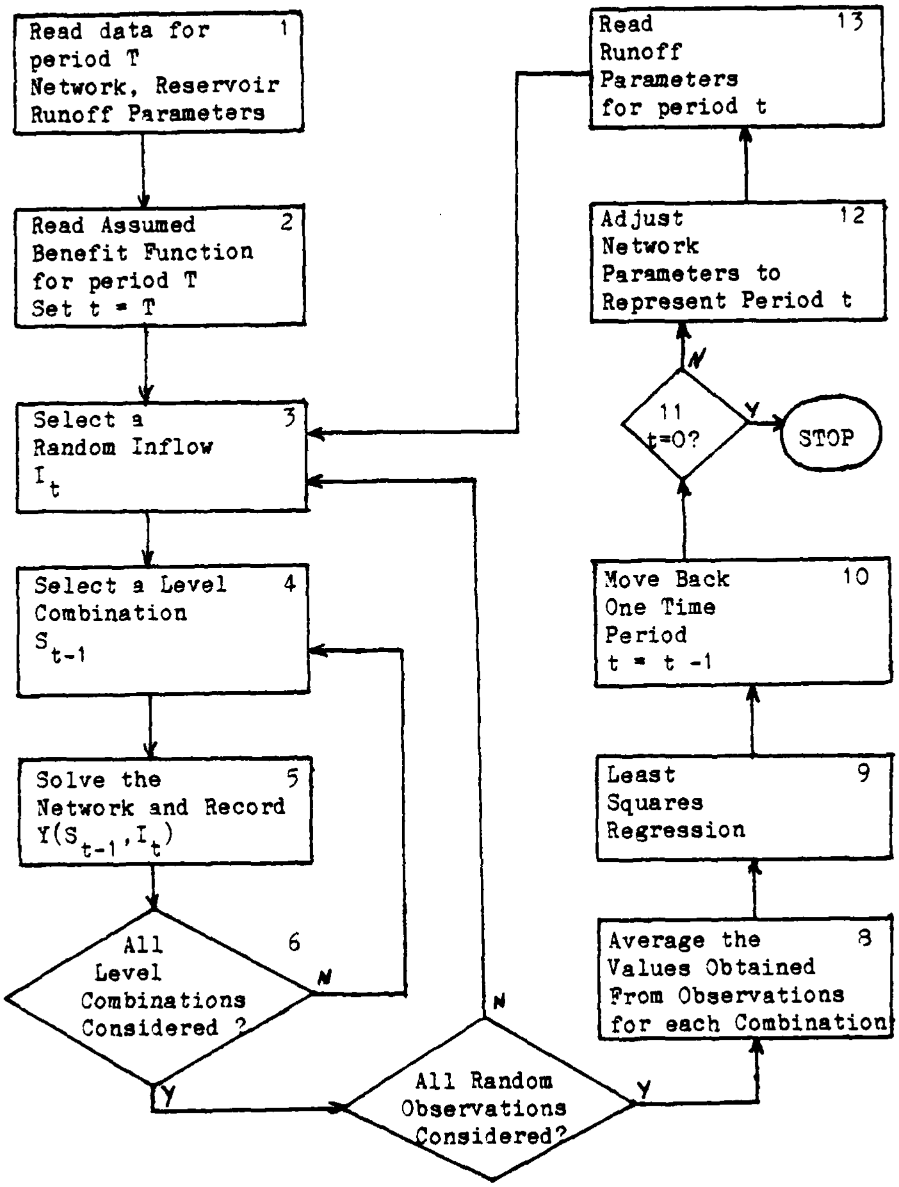

Read data for period T Network, Reservoir Runoff Parameters 1
Read Assumed Benefit Function for period T Set t = T 2
Select a Random Inflow I_t 3
Select a Level Combination S_{t-1} 4
Solve the Network and Record $Y(S_{t-1}, I_t)$ 5
6 All Level Combinations Considered ?
N
Y
All Random Observations Considered?
N
Y
Average the Values Obtained From Observations for each Combination 8
Least Squares Regression 9
Move Back One Time Period t = t -1 10
11 t=0?
Y
STOP
N
Adjust Network Parameters to Represent Period t 12
Read Runoff Parameters for period t 13

Figure 4-3

Dynamic Programming Approach

percent of CK_r. There will be NN components of RL.

ZZ = Value of the random number

L_R = Number of level combinations given R reservoirs. Thus,

$$L_R = NN^R$$

K = Number of random observations per level combination of the vector I_t taken from the distribution of I_t

$BF(S_t)$ = Expected benefit for water stored at the end of period t for t = 1,...,T

$Y(S_{t-1},I_t)$ = Model response for level combination S_{t-1} when the random inflow is I_t

Using the above notation some sample values will be assumed for purposes of illustrating the overall process. For this example, let:

$R = 2$ reservoirs

$NN = 3$

$CK_1 = 20$, $CK_2 = 40$

$RL = (.5, .7, .9)$

$K = 10$

$BF(S_t) = 20f_1 + 25f_2 - f_1^2 - f_2^2 - .45f_1f_2$

Then:

$L_R = 3^2 = 9$ level combinations

These 9 level combinations will be represented by the vector $S_{t-1} = (s_{1t}, s_{2t})$. The possibilities are listed in Table 4-1.

Table 4-1

Level Combinations for the Example

s_{1t}	s_{2t}
10	20
10	28
10	36
14	20
14	28
14	36
18	20
18	28
18	36

This example will be continued later.

The steps of the algorithm corresponding to the boxes of the flow chart in Figure 4-3 are as follows:

1. Read the network and reservoir data. This includes reading all of the network arc parameters and all reservoir data (NN, CK_r, W, RL, T, etc.). Also read the runoff parameters for period T for each reservoir. These define the distribution of random inflows I_t. Set t = T.
2. Read $BF(S_T)$: This is the assumed functional representation for the future value of water at the end of period T. This information relates to the cost parameters to be assigned to the nonlinear arcs for period T. For the two reservoir example, read two linear coefficients two coefficients for

the squared terms and one coefficient for the cross product.

3. Draw a random observation from the distribution I_t.
4. Select a level combination. From the L_R combinations, select one that has not been evaluated. This level combination is the vector S_{t-1} since it represents a given initial set of reservoir contents for period t. Go to 5.
5. Solve the network. Since $BF(S_t)$, S_{t-1} and I_t are known, the single period network problem is deterministic.

   Let the optimal solution to the network be given as:

   $$Y(S_{t-1},I_t) = Min(HF_t - BF(S_t))$$

   subject to conservation of flow and bounding constraints. This is not a linear problem since $BF(S_t)$ is nonlinear and S_t contains variables of the flow problem. The problem is a nonlinear network flow problem. The solution procedure for this nonlinear network will be discussed in Chapter 5. To further simplify the notation, let:

   $$Y_{i,w} = Y(S_{t-1},I_t) \quad w = 1,\dots,K$$

   This is equivalent to stating that there are now i different level combinations and for each of these level combinations K random draws will be made. Thus, $Y_{i,w}$ represents the value of the network optimal solution for the i^{th} level and the w^{th} draw. Go to 6
6. If all level combinations have not been evaluated, go to 4. Otherwise, L_R values of $Y_{i,w}$ will have been generated for

this choice of w, one for each level combination. For each of these level combinations the same random inflows, I_t, will have been added. Go to 7.

7. If all random draws have not been made, go to 3. Otherwise, for each random draw all level combinations will have been evaluated. A total of L_R times K (90 for this example) optimal solutions will have been generated. Each random inflow I_t was applied to all L_R combinations. This aspect of the algorithm is discussed more fully in section 4.6. Go to 8.

8. Average the values obtained from the observations for each level combination. Go to 9.

9. Least squares regression. Perform a least squares regression using the averages of the observations as the dependent variable. This will be discussed more fully in section 4.7. Go to 10.

10. Move back one time period: Let t = t-1. Go to 11.

11. If t = zero, STOP. Otherwise, go to 11.

12. Adjust network parameters. This requires adjusting the cost parameters for the nonlinear arcs. These new parameters will be the coefficents of the derived benefit function. Go to 12.

13. Read runoff parameters for period t-1. Go to 3.

Once $BF(S_{T-1})$ is available, then $BF(S_{T-2})$ is computed in a like manner. The process continues until $BF(S_1)$ is evaluated. This approach is different than the classical dynamic programming approach in the following ways:

1. The benefit function is represented as a continuous mathematical function rather than for discrete values of the state variables.

2. The recursive equation is solved using a Monte Carlo sampling approach rather than using transition probabilities.

3. The optimum decisions are found using a network flow algorithm rather than a discrete search over a finite set of decisions.

The solution procedure of the network optimization algorithm requires that the form of the benefit function be specified. This is necessary since a different solution technique would be required for solving the nonlinear form of the benefit function. While any convex form could be applied, the next section discusses the rational for using a quadratic to represent these benefit functions.

4.6 Sampling From the Distribution of Inflows

An important aspect of the algorithm is the derivation of the expected benefit by sampling from the distribution of inflows. The runoff distributions provided as an input to the procedure may take many forms. These may include normal, log-normal, log-normal

Pearson Type III, exponential, historical, etc. The computer program currently has three options available, normal, log-normal and historical. Other inflow distributions could easily be incorporated.

As an example of how these inflow parameters are used, consider again the example problem used earlier. If the distribution of runoff is assumed to be normal, the mean and standard deviation for inflows will be read for each reservoir in box 1 of Figure 4-3. Let the mean and standard deviation for the two reservoirs be as follows:

Reservoir 1 Mean = 5, Standard Deviation = 2

Reservoir 2 Mean = 8, Standard Deviation = 3

Next, in box 3, a random number ZZ will be generated from a normal distribution with a mean of zero and a standard deviation of 1. ZZ is used in conjunction with the mean and standard deviation for each reservoir. Thus, the total inflow for each reservoir will be equal to:

$$I_t = \text{Mean} + ZZ * \text{Std. Dev.}$$

If I_t is less than or equal to zero, a zero inflow is assumed. For example, if ZZ = .5, $i_{1t} = 5 + 1 = 6$, and $i_{2t} = 8 + 1.5 = 9.5$.

This implies that for the model developed, perfect correlation of inflows between the reservoirs in a basin is assumed. This is not a requirement of the algorithm and other assumptions could be made.

In addition to applying the random number ZZ to all reservoirs, this same random inflow will be applied to all reservoirs for all level combinations. For each of the 9 level combinations of the example, add 6 and 9.5 units to the initial contents for reservoirs 1 and 2 respectively. This will result in total water available for level combination 1 of 16 for reservoir 1 and 29.5 for reservoir 2. For the second level combination, these values would be 16 and 37.5, etc. This process is used to assure convexity of the response surface. This assures that if the first level combination is overestimated, all of the remaining level combinations will also be overestimated. To minimize this over (or under) estimation it is required that a sufficient number of random draws be made to make this error negligible. This will be discussed more fully in Chapter 6.

4.7 Least Squares Regression

As indicated by the algorithm a least squares regression is performed on the mean responses of the L_R level combinatons. If all level combinations have been evaluated, there will exist a matrix of optimum solution values referred to here as the response matrix as illustrated in Figure 4-4. The rows of the response matrix represent the L_R level combinations and the K columns represent the network optimal solutions, $Y_{i,w}$, for each of the K draws. Preceding this response matrix is another matrix which will be referred to as the design matrix. This matrix will have L_R rows and U columns, where U depends upon the number of terms in the

Design Matrix

Response Matrix

U Terms of
Benefit Function

Random Draws

Mean
Response

$$
\begin{array}{c|cccc|ccccccc|c}
 & & & & & 1 & 2 & \cdot\;\cdot & w & \cdot\;\cdot\;\cdot & K & & \\
\hline
1 & 1 & f_{11} & f_{12} & f_{13}\;\cdot\;\cdot & Y_{11} & Y_{12} & \cdot\;\cdot & Y_{1w} & \cdot\;\cdot\;\cdot & Y_{1K} & & \overline{Y}_1 \\
2 & & & & & & & & & & & & \cdot \\
\cdot & & & & & & & & & & & & \cdot \\
\cdot & & & & & & & & & & & & \cdot \\
i & 1 & f_{i1} & f_{i2} & & Y_{i1} & \cdot\;\cdot\;\cdot\;\cdot & & Y_{iw} & \cdot\;\cdot\;\cdot & Y_{ik} & & \overline{Y}_i \\
\cdot & & & & & & & & & & & & \cdot \\
\cdot & & & & & & & & & & & & \cdot \\
\cdot & & & & & & & & & & & & \cdot \\
L_R & 1 & f_{L_R1} & f_{L_R2}\;\cdot\;\cdot & & Y_{L_R1} & \cdot\;\cdot\;\cdot & & Y_{L_Rw} & \cdot\;\cdot & Y_{L_RK} & & \overline{Y}_{L_R} \\
\hline
\end{array}
$$

Level Combinations

Independent
Variables

Dependent
Variables

where

$$\overline{Y}_i = \frac{1}{K}\sum_{w=1}^{K} Y_{iw}$$

Figure 4-4

Matrices For Least Squares Regression

benefit function $BF(S_{t-1})$. If it is desired to fit a linear benefit function to this data in terms of the R reservoirs, $U = R + 1$, (R linear terms plus a constant). To represent the benefit function as a quadratic, linear, second order and interactive terms are required. Thus, for the quadratic, $U = 2 * R + C_2^R + 1$. Thus, the U terms in each row represent the desired form of the benefit function. For the example, $U = 6$.

To fit the desired form of the benefit function to this data, a least squares regression is performed. Here, the design matrix has as its first column, a vector of ones. This is necessary to account for the constant term. Accordingly, this design matrix represents the independent variables for he regression analysis. As the dependent variable, the means of the rows of the response matrix will be used. Thus, the dependent variable for row i is:

$$\overline{Y}_i = \frac{1}{K}\sum_{w=1}^{K} Y_{i,w} \quad \text{for all } i$$

By fitting this data to the selected design, $BF(S_{t-1})$ is derived.

4.8 The Quadratic Benefit Function

The usual procedure of discrete dynamic programming is to store the return function $f(S_t)$ in computer memory for a large number of discrete values of the state variables S_t. When S_t is a multidimensional vector (one dimension for each reservoir) the number of possible discrete states can be very large. Because rapid access computer memory is finite in size, this limits the number of state variables that can be handled at each stage. Three state variables is frequently described as a practical limit.

In the method presented here, this storage problem is overcome by fitting a quadratic function to $BF(S_t)$ and storing only the coefficients of the quadratic.

A quadratic form has been chosen for the following reasons:

1. It can exhibit the concave shape expected for the benefit function. (convex cost function)
2. It can represent nonseparable interactions between reservoirs with cross product terms.
3. It is easy to store in a computer.
4. It is computationally convenient in the network models which arise in the solution procedure.

If S_t has n dimensions, the number of terms in a quadratic is:

$$\frac{1}{2}(n^2+3n)+1$$

For the three reservoir case the full three variable quadratic would have 10 terms as shown here:

$$B_0+B_1f_1+B_2f_2+B_3f_3+B_4f_1{}^2+B_5f_2{}^2+B_6f_3{}^2+B_7f_1f_2+B_8f_1f_3+B_9f_2f_3$$

Storing these 10 coefficients is much easier than storing the NN^R discrete state vectors.

Also beneficial is the fact that the function is defined for continuous S_t rather than discrete values. This means that once such a function has been derived (albeit from an approximation to the discrete representation) decisions can be made by a one time solution of the network using the observed reservoir levels. These reservoir levels need not be equated or rounded to the nearest discrete level.

It is clear that the true expected benefit function is not in reality a quadratic function. A better fit to the observed data might be obtained with a more complex model. This research has limited consideration to the quadratic because of the reasons noted above. The dynamic programming methodology however is not limited to this case. Indeed the data could be fit to any model and the recursive procedure is independent of the model. The network flow solution procedure is limited to the quadratic case, however it probably would be possible to derive a more general procedure along the lines of nonlinear algorithms for pure network flow problems (Luenberger (1965)). This was outside the scope of this research.

The next chapter will present the methodology for solution of the nonlinear (quadratic) network. The method for obtaining data for use in the least squares program will be fully explained in Chapter 6.

Chapter V

5. Solution of Nonlnear Network Problems

5.1 Introduction

In the performance of the dynamic programming algorithm it is necessary to solve network problems with convex, quadratic, nonseparable arc costs. In Figure 4-1 the flow on arcs 1, 2 and 3 represent water stored in reservoirs for future use. The benefit function for this water stored is represented as a concave, quadratic function. As an example of a quadratic concave benefit function, $BF(f_1,f_2,f_3)$ is assumed to be:

$$59f_1+46f_2+39f_3-.86f_1^2-.53f_2^2-.52f_3^2-.64f_1f_2-.40f_1f_3-.68f_2f_3$$

where f_1, f_2 and f_3 are the flows on arcs 1, 2 and 3 respectively. This benefit function will be used later as the period T assumed benefit function for many of the example problems of Chapter 7. Since the algorithm operates on costs the negative of the benefit function is used to obtain the cost function. Thus:

$$C(f_1,f_2,f_3) = -BF(f_1,f_2,f_3)$$

This is a convex cost function. The network flow programming algorithms which have appeared in the literature (Ali et al. (1978), Cooper and Kennington (1977), Dembo and Klincewicz (1979), Florian (1977), Frank and Wolfe (1956), Helgason and Kennington (1978) and Klincewicz (1979) have been designed for linear or convex problems, but not for generalized problems with nonseparable objective functions. This chapter provides details on the theoretical development of an algorithm to handle quadratic nonseparable objective functions. The procedure is designed for generalized networks (i.e. networks with gains) with convex, quadratic, nonseparable objective functions and has been coded in Fortran for the CDC computer.

5.2 Problem Statement

Consider a network problem defined as in Chapter III with the added stipulation that a subset of the arcs, M_N, have nonlinear arc costs. This subset is included in the set of all arcs M. The linear cost coefficients are described by the vector H for all arcs. A matrix will be used to define the nonlinear component of cost.

The cost of nonlinear arcs is assumed to be a quadratic function of arc flows. Let F_N be the vector of flows in the nonlinear arcs and Z_N be the nonlinear cost contribution of the nonlinear arcs. Then:

$$Z_N = F_N^T \, Q \, F_N$$

where Q is a symmetric matrix which is positive definite and the T represents the transpose of a vector.

For the example problem:

$$Q = \begin{bmatrix} .86 & .32 & .20 \\ .31 & .53 & .34 \\ .20 & .34 & .52 \end{bmatrix}$$

This is a positive definite matrix, so Z is a convex function. This Q matrix is equivalent to the Hessian of $C(f_1,f_2,f_3)$ divided by 2. This notation is common in the literature and Q is referred to as the quadratic matrix.

The total cost for the system flow is:

$$Z=HF+F_N^T\ Q\ F_N$$

The nonlinear arcs are also represented in the flow vector F so that linear costs can also be associated with the arcs.

Define H' to be the vector of first derivatives of the arc costs. Of course, for the linear arcs:

Eq. (1):

$$h'_k=h_k$$

For nonlinear arcs:

Eq. (2):

$$h'_k=h_k+2Q_kF_N$$

where Q_k is the kth row of Q. In this chapter, Q will be subscripted with k to indicate a specific row in the Q matrix. When Q is not subscripted, it will refer to the entire matrix. In Chapter 6, the matrix Q will be subscripted as Q_t when it is desired to identify it with a specific time period.

The algorithm has to find a minimum cost solution for the flow vector F (which includes the nonlinear flows F_N). The nonlinear network model is:

Model III

$$\text{Minimize } HF + F_N^T Q F_N$$

st.

$$\sum_{k \in M_{Oi}} f_k - \sum_{k \in M_{Ti}} a_k f_k = b_i \quad i = 1,\ldots,n-1$$

$$0 \leq f_k \leq c_k$$

$$k = 1,\ldots,M$$

These are the same network constraints as for the linear model. A primal approach is used in which an initial basic feasible solution is defined. This solution describes a basis network. A basis network for the pure network is a set of n-1 arcs which form a tree rooted at the slack node and having a directed path from the slack node to all the nodes of the network. For the generalized network, the basis network may consist of several components, one of which is a tree rooted at the slack node and the other are trees rooted at cycles. A component consists of a set of nodes such that there is a path between every pair of nodes in the set.

The solution procedure for the nonlinear network parallels that of the primal linear solution algorithm of Chapter 3.3. This algorithm is restated here.

1. Check each nonbasic arc for complementary slackness,

$$\text{if } (\pi_i + h_k')/a_k < \pi_j \text{ then } f_k = c_k$$

$$\text{if } (\pi_i + h'_k)/a_k > \pi_j \text{ then } f_k = 0$$

If each nonbasic arc does not violate either of these conditions, stop, the solution is optimal. Otherwise, choose an arc to enter the basis that violates one of these conditions. Let this be arc k_E. Note that h'_k is used in these conditions for the nonlinear problem.

2. For each arc in the basis, find the amount of flow change in the arc per unit of flow change in arc k_E. Use this information to find the maximum flow change in arc k_E that will cause the flow in one of the basic arcs to go to a bound or cause the flow in arc k_E to go to its opposite bound. Choose the arc to leave the basis, k_L, as the arc which limits the flow.

3. Change the flow in arc k_E and the basis arcs by the amount found in step 2. If $k_L = k_E$, return to step 1. Otherwise, change the basis tree by deleting arc k_L and inserting arc k_E. Change the node potentials to be consistent with the new basis network. Return to step 1.

For a linear problem the iterative step of the primal method checks all nonbasic arcs for optimality. This is the test for complementary slackness (Chapter III) which evaluates:

$$d_{k_E} = \pi_{i_E} + h'_{k_E} - a_{k_E}\pi_{j_E}$$

where the subscript E refers to the entering arc. If d_{k_E} is less than zero and f_{k_E} is zero, then the network is not optimal and arc k_E will attempt to enter the basis. d_{k_E} is interpreted as the change in the objective function per unit change of flow in arc k_E. Since d_{k_E} is negative, and the model is trying to minimize costs, it would like to put as much flow on this arc as possible. For each unit of flow added to arc k_E, the objective function will change by d_{k_E}. In this case the objective function will be reduced since $d_{k_E} < 0$. The other nonoptimal condition is where $d_{k_E} > 0$ and $f_{k_E} = c_{k_E}$. In this case the algorithm will remove flow from arc k_E, and for each unit removed, the objective function will be decreased by the amount d_{k_E}. Because of the conservation of flow conditions, changing flow on arc k_E means adjusting the flows on other basic arcs. The final result is that eventually one of these arcs, either k_E or a basic arc, will reach its bound on flow and no further improvement can be made. If there is enough slack in the network to allow arc k_E to receive as much flow as it can handle (i.e. f_{k_E} goes to one of its bounds) then arc k_E is not allowed to enter the basis rather it becomes nonbasic at the opposite bound. However, if there is a basic arc whose flow goes to one of its bounds before the flow on the entering arc reaches a bound this basic arc will leave the basis and arc k_E will enter.

There are two principle differences between the nonlinear and linear networks which require special consideration in the algorithm. The first is that as flow changes on the nonlinear arcs, their marginal costs, h_k', will also change. These costs are a linear function of the flows and are not constants. Thus a flow change on the basic arcs and arc k_E may simultaneously change the dual variable, π, and the value of h'_{k_E}. This results in the second difference. A flow smaller than the one required to cause either k_E or a basic arc to reach a bound may cause the entering arc to satisfy complementary slackness. Thus, it may be that a nonlinear arc will remain nonbasic, even though the flow on the nonbasic arc is between its bounds. For the linear problem each nonbasic arc will have flow at zero or capacity, while for the nonlinear problem a nonbasic flow may be between zero and capacity. These differences will be illustrated below by first demonstrating the algorithm with a linear network and then showing the differences with a nonlinear network.

Assume that in step 1 of the primal simplex algorithm, a nonbasic arc is discovered which violates complementary slackness. For this discussion, let $d_{k_E} < 0$. In the linear network this would imply that $f_{k_E} = 0$. The networks of Figure 5-1 illustrate the steps of the primal simplex algorithm for a pure linear network ($a_k = 1$). The basis for Figure 5-1a is made up of arcs (2,3,4). The node potentials are determined by the basic arcs. Thus, $\pi_3 = \pi_1 + h_2 = 0 + 2 = 2$. $\pi_2 = \pi_3 + h_3 = 2 + 3 = 5$ and $\pi_4 = \pi_2 + h_3 = 5 + 1 = 6$. Step 1 of the algorithm checks each nonbasic arc for complementary slackness. The nonbasic arcs are arcs 1 and 5. For arc 1, $d_1 = \pi_1 + h_1 - \pi_2 = -3$ and $d_5 = \pi_3 + h_5 - \pi_4 = -1$. Both arcs 1 and 5 have $d_k < 0$. Arc 5, however, is at capacity ($f_5 = c_5$) which means that complementary slackness is satisfied. Arc 1 has zero flow and a negative d_k implies a nonoptimal condition. Arc 1

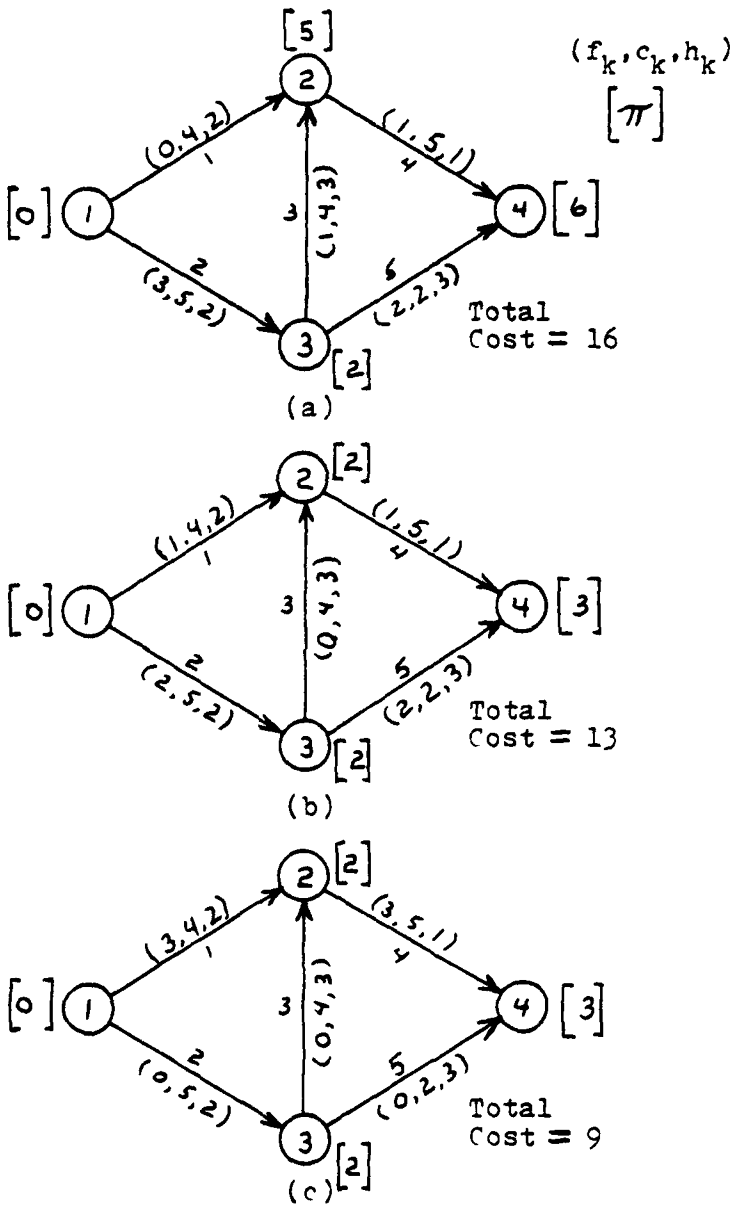


Figure 5-1

Primal Simplex Algorithm for Linear Network is identified as the entering arc, $k_E = 1$.

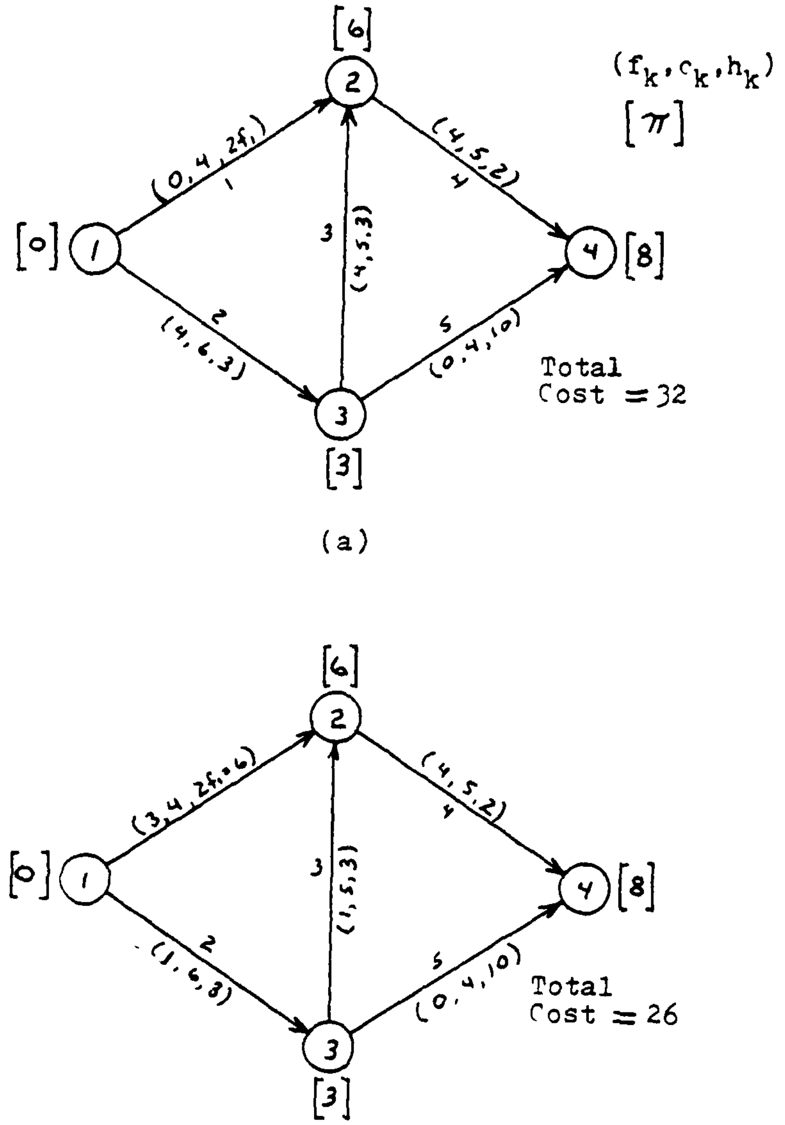
Step 2 of the algorithm finds the arc which will limit the flow change as flow is changed on arc k_E. As flow is added to arc 1, flow must be decreased on basic arcs 2 and 3 to maintain conservation of flow. A decrease of 1 unit on arc 3 results in a flow of zero on this arc. This is the limiting arc and $k_L = 3$. Step 3 of the algorithm changes the flows according to the step 2 results and updates the node potentials. Figure 5-1b shows the results of this iteration of the algorithm. The basic arcs are now (1,2,4). The total cost for the network flows went from 16 to 13 for this iteration. d_{k_E} is the change in total cost for a unit change of flow in arc k_E. d_{k_E} was -3 and one unit of flow was added to arc $k_E = 1$. The total cost changed by $16 - 3 = 13$. The algorithm now returns to step 1 to check for optimality.

The basis for Figure 5-1b includes arcs (1,2,4). Evaluating the nonbasic arcs (3,5) results in $d_3 = 2 + 3 - 2 = 3$ and $d_5 = 2 + 3 - 3 = 2$. Arc 3 has zero flow and complementary slackness is satisfied. Arc 5 has $f_5 = c_5$ and $d_5 > 0$. This fails the test for complementary slackness and $k_E = 5$. In this case, flow will be removed from arc 5 and arc 2 and added to arcs 1 and 4. As flow is removed, the flow on arcs 2 and 5 both reduce to zero before the flow on arc 1 or 4 reach their upper bound. In this case, arc 5 is considered the limiting arc and $k_E = k_L = 5$. The resulting flows and π updates are shown in Figure 5-1c.

The basis for Figure 5-1c remains the same as Figure 5-1b

(1,2,4), with a total cost of 9. For $k_E = 5$, $d_{k_E} = 2$ which implies that for each unit change of flow on arc 5, the total cost will decrease by 2. For a change of 2 units, total cost becomes 13 - 2(2) = 9. Returning to step 1 of the algorithm reveals that the flows on the network of Figure 5-1c are optimal.

Figures 5-2 and 5-3 are intended to show the differences between linear and nonlinear networks with respect to the steps of the algorithm. The network of Figure 5-2 has the same form as Figure 5-1 but different arc parameters. Arc 1 is now a nonlinear arc with an arc cost function of $h_1 = f_1^2$. This arc is presently nonbasic with zero flow as shown in Figure 5-2a. The basis includes arcs (2,3,4). Step 1 of the algorithm evaluates the nonbasic arcs for complementary slackness. $d_5 = 3 + 10 - 8 = 5$ and complementary slackness is satisfied. Arc 1 however, has $d_1 = 0 + 2f_1 - 6 = -6$ since $f_1 = 0$. Thus, $k_E = 1$. Step 2 of the algorithm finds the arc which limits the flow change. As flow is increased on arc 1 flow must be decreased on arcs 2 and 3. If the network of Figure 5-2 were linear, it would be profitable to add 4 units of flow to arc 1 and remove 4 units from arcs 2 and 3, causing all three of these arcs to go to a bound, in which case any one of these arcs could leave the basis. For $d_1 = -6$, this would reduce the total cost to 8 since $h_1 = 0$. The marginal cost for arc 1 will not remain zero as flow is added for the nonlinear case. Consequently, it is necessary to find the amount of flow which when added to arc 1 will cause d_1 to go to zero. In this case, for f_1 =



(a)

(b)

Figure 5-2

Nonlinear Network Algorithm Differences

3, $h_1' = 6$ and $d_1 = 0 + 6 - 6 = 0$. Thus, the flow change is limited to 3 on the nonlinear arc. Comparing this limiting flow with the limiting flow for the linear arcs of 4 and choosing the smallest results in arc 1 being the limiting arc.

Figure 5-2b shows the results of this flow change and the updated node potentials. Several things should be noted in Figure 5-2b. First, as in the linear network, if k_E is the limiting arc, $k_L = k_E$. Thus, the basis of Figure 5-2b is the same as Figure 5-2a. However, flow on arc 1 (a nonbasic nonlinear arc) is not at a bound. This is allowed for nonlinear networks but not for linear networks.

Secondly, when evaluating the d_k for the nonbasic arcs of Figure 5-2b, $d_1 = 0$. This is due to the previous iteration which forced this result. Evaluating arc 5, yields $d_5 = 5$ and the network of Figure 5-2b is optimal. The total cost is now 23.

A third feature to note is that the total cost did not reduce to 8 as it would have if this were a linear network. Thus, for the nonlinear network, the evaluation of d_k is necessary to test for complementary slackness, but it cannot be interpreted as the change in total cost per unit change of flow in arc k_E for nonlinear networks, rather it is the derivative of this change.

In the example above, the node potentials did not change since the basis remained unchanged and contained only linear arcs. If the nonlinear arc had entered the basis and since there are no other nonlinear arcs in the basis, the node potentials would be updated in the usual manner, that being only to update the node potentials in the subtree rooted at the terminal node of the entering arc. For a more complicated network there may be other nonlinear arcs in the basis. These arcs may or may not be in the subtree rooted at the terminal node of the entering arc. If they are, the node potentials will be updated routinely. Basic nonlinear arcs in another part of the network may not experience a flow change, however, if their costs functions include an interactive term relating to an arc whose flow did change, then the marginal cost associated with the nonlinear arc will change. This requires that the node potentials be updated for all nodes in all subtrees rooted at the origin of the nonlinear arcs in the basis.

Consider the simple example of Figure 5-3. There are two nonlinear arcs in this network, arcs 1 and 2. Let the cost associated with arc 1 be $h_1(f_N) = f_1 + .5f_1f_2$ and further suppose that the flow on arc 2 is zero. Consequently, the node potentials for nodes a,b, and c are (1,11,16) respectively. Let $k_E = 2$ enter the basis with a flow of 2 and suppose k_L = arc 3. The node potentials for nodes d,e and f would be updated in the usual way since they are in the subtree rooted at the terminal node of the entering arc. For the linear problem, these are the only node potentials that would require updating. In the nonlinear case, since the cost of arc 1 is a function of the flow on both arcs 1 and 2, the marginal cost for arc 1 will change.

$$h_1' = f_1 + .5f_2$$

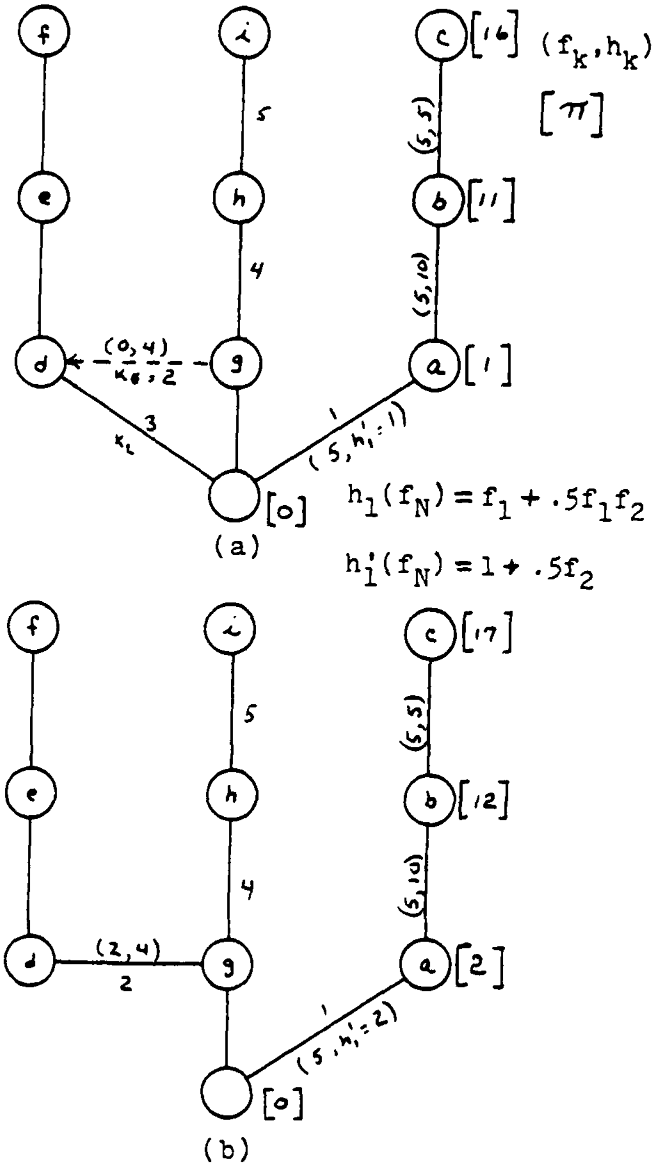


Figure 5-3

Nonlinear Arc Cost Changes

$$= 1 + .5(2) = 2$$

Thus, the marginal cost now associated with arc 1 is 2 instead of 1. Accordingly, the node potentials for nodes a,b and c will require updating to values of (2,12,17) respectively. This is unique to the nonlinear network and must be accounted for in the algorithms. The node potentials for nodes h and i will remain unchanged if arcs 4 and 5 have linear arc cost functions.

The main changes to the linear network codes to allow use of the primal simplex algorithm for the nonlinear problem involve three primary areas:

1. Determining the arc costs and node potentials
2. Calculating the effect of flow change
3. Finding the value of flow change which causes i_{k_E} to go to zero for a nonlinear arc k_E.

These three issues will be addressed in the remaining three sections of this chapter.

5.3 Arc Cost and Node Potential

The dual variable π_i represents the marginal cost of increasing flow to node i. For the pure linear problem:

Eq. (3):

$$\pi_j = \pi_i + h'_k \quad k(i,j) \in M_B$$

Setting the node potential for the slack node equal to zero allows the determination of the node potential for all other nodes in the network. For the generalized linear problem:

Eq. (4):

$$\pi_j = \frac{\pi_i + h'_k}{a_k} \quad k(i,j) \in M_B$$

For the linear problem the marginal cost of flow through an arc is constant. For the nonlinear network, the marginal cost is not constant but is a function of flow. The marginal cost h'_k is the partial derivative of the arc cost function with respect to f_k. If nodes i and j happen to be on a cycle a more complex equation is required for calculating the node potentials. Let M_c = $(1,2,\ldots,k_c)$ represent the nodes of a cycle. The π value at node 1 is:

Eq. (5):

$$\pi_i = \frac{h'_1 + \sum_{k=2}^{k_c} h'_k \prod_{j=1}^{k-1} a_j}{B-1}$$

where B is the cycle gain, defined as the product of the arc gains on the cycle.

$$B = \sum_{i=1}^{k_c} a_i$$

The numerator of equation 5 is referred to as the unit cost of the cycle. Once one of the node potentials for a node on the cycle is determined, the remaining can be calculated using equation 4.

From equation 2 the marginal cost for linear arcs is just h_k and for the nonlinear arcs in a quadratic model is:

Eq. (6):

$$h_k' = h_k + 2Q_k F_N$$

Thus, for the nonlinear problem:

Eq. (7):

$$\pi_j = \frac{\pi_i + h_k'}{a_k} \qquad \text{for } (k(i,j) \in M_N \subset M_B$$

5.4 The Effect of Flow Changes

When an arc is to enter the basis, flow is changed in the basic arcs. To determine what these flow changes are, a quantity Y_j is computed. Y_j represents the flow change through node j, for a unit flow change in the entering arc.

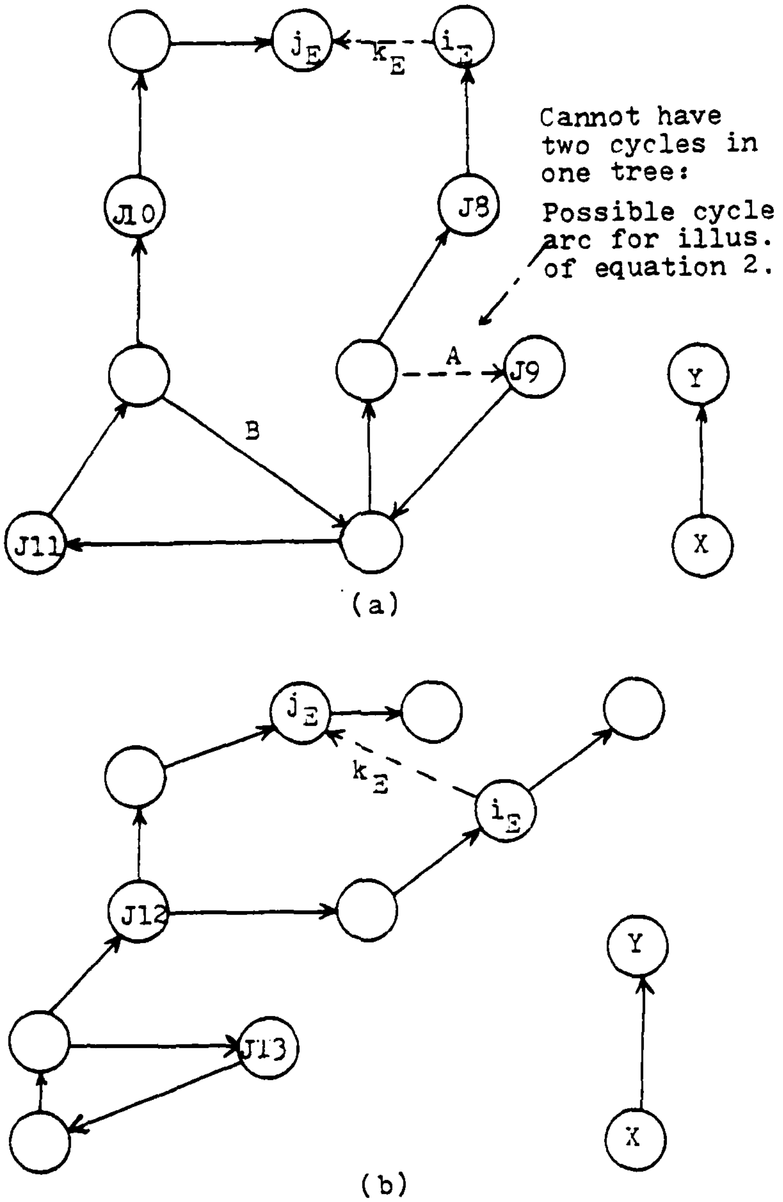
The equations which are presented here without justification are described in detail in Jensen and Barnes (1980). It is important to present them in this form because in a nonlinear problem the flow in basic arcs effects the node potentials and hence the value of d_{k_E}. As stated above, it is possible that a flow change will drive d_{k_E} to zero before an arc is driven out of the basis and before the flow on arc k_E reaches its bound.

Assume that an arc $k_E(i_E,j_E)$ is chosen to enter the basis such that:

$$d_{k_E} = \pi_{i_E} + h_{k_E} - a_{k_E}\pi_{j_E} < 0 .$$

The quantity γ_j is the amount of flow change through node j for each unit change of flow in arc k_E. Depending upon the current configuration of the basis, and on the location of node j within this configuration, various equations apply for calculating γ_j..

To facilitate understanding of these equations, reference is made to Figure 5-4a and 5-4b. The key to the various equations lies in the positioning of node j in the basis network. If the basis contains a cycle, the basis tree will be broken into two parts or semitrees as shown in Figure 5-4. Assume that the trees and semitrees of Figure 5-4 represent a basis for some 12 node network. In Figure 5-4a arcs A and B cannot both be in the basis at the same time since there cannot be two cycles in the same component of the basis network. However, for illustrative purposes, both are shown here. The nodes indicated as i_E and j_E represent the origin and terminal nodes for the entering arc respectively. The node notation Jn in the figure refers to the node in the basis which corresponds to the equation numbers below:



Cannot have two cycles in one tree:

Possible cycle arc for illus. of equation 2.

(a)

(b)

Figure 5-4
Basis Trees

Eq. (8): When node j is on the basis path to node i_E but not on a path to j_E nor on a cycle:

$$\gamma_j = \frac{1}{\prod_{k \in P_{ji_E}} a_k}$$

where P_{ji_E} is the directed path from node j to node i_E defined by the basis. For a basis network this path is always unique. Node J8 in Figure 5-4a is an example of a node for which equation (8) is appropriate.

Eq. (9): If node j is on the basis path to node i_E, not on a path to j_E, but lies on a cycle:

$$\gamma_j = \frac{\frac{(B-1)}{B}}{\prod_{k \in P_{ji_E}} a_K}$$

where B is the cycle gain. To illustrate equation 9 in Figure 5-4a, remove arc B from the basis and add arc A. Now, J9 is a node corresponding to equation (9). For equations (10) and (11) refer to the original basis.

Eq. (10): For nodes such that there exists a directed path from node j to node j_E (using basic arcs) but not to i_E and node j is not on a cycle:

$$\gamma_j = \frac{-a_{k_E}}{\prod_{k \in P_{jj_E}} a_k}$$

where P_{jj_E} is the set of arcs on the path from node j to node j_E.

Eq. (11): For nodes such that there is a directed path from node j to node j_E (using basic arcs) but not to i_E and node j is on a cycle:

$$\gamma_j = \frac{-\frac{(B-1)}{B} a_{k_E}}{\prod_{k \in P_{jj_E}} a_k}$$

Eq. (12): If there exists a directed path from node j to both i_E and j_E, and j is not on a cycle:

$$\gamma_j = \frac{1}{\prod_{k \in P_{ji_E}} a_k} - \frac{a_{k_E}}{\prod_{k \in P_{jj_E}} a_k}$$

This is equal to equation (8) plus equation (10).

Eq. (13): If there exists a directed path from node j to both i_E and j_E, and j happens to be on a cycle:

$$\gamma_j = \frac{(B-1)}{B} \frac{1}{\prod\limits_{k \in P_{ji_E}} a_k} \qquad \frac{a_{k_E}}{\prod\limits_{k \in P_{jj_E}} a_k}$$

This is equal to equation (9) plus equation (10).

Nodes that do not lie on a directed path to either i_E or j_E have:

$$\gamma_j = 0$$

This is represented by nodes X and Y in Figures 5-4a and 5-4b. Note that a positive value of γ_j indicates the flow in basic arc k(i,j) will increase, and a negative value indicates that this flow will decrease.

If arc k(i,j) is a member of the basis the flow change in arc k per unit increase in f_{k_E} is:

Eq. (15):

$$\frac{\gamma_j}{a_k}$$

For convenience define the quantity:

$$g_k = \frac{\gamma_j}{a_k}$$

$$k(i,j) \in M_B$$

The maximum flow increase for arc k_E is that value that will cause a basic arc (or arc k_E) to go to a bound. This maximum is:

Eq. (16):

$$\nabla^{+} = \text{Min}\left[(c_{k_E} - f_{k_E}), \right.$$
$$\text{Min}\left[(c_k - f_k)\ g_k \quad k(i,j) \in M_B,\ g_k > 0\right],$$
$$\left. \text{Min}\left[-f_k\ g_k \quad k(i,j) \in M_B,\ g_k < 0\right]\right]$$

This equation is used if $d_{k_E} < 0$ and $f_{k_E} < c_{k_E}$. The maximum flow decrease in arc k_E is:

Eq. (17):

$$\nabla^{-} = \text{Min}\left[f_k, \right.$$
$$\text{Min}\left[f_k\ g_k \quad k(i,j) \in M_B,\ g_k > 0\right],$$
$$\left. \text{Min}\left[-(c_k - f_k)\ g_k \quad k(i,j) \in M_B,\ g_k < 0\right]\right]$$

This equation is used if $d_{k_E} > 0$ and $f_{k_E} > 0$. Equations (16) and (17) are used to determine the flow change which will drive an arc flow to a bound. For the nonlinear problem it is also necessary to determine the flow change that will cause d_{k_E} to go to zero. This is the subject of the next section.

5.5 Flow Change Which Drives d_{k_E} to Zero

For the nonlinear problem the value of d_{k_E} is determined by:

Eq. (18)

$$d_{k_E} = \pi_{i_E} + h'_{k_E} - a_{k_E}\pi_{j_E}$$

Assume $f_{k_E} < c_{k_E}$ and $d_{k_E} < 0$ indicating a nonoptimal condition. The results of the last section indicate the flow change that causes a basic arc (or arc k_E) to go to a bound. Since π_{i_E}, h'_{k_E} and π_{j_E} may all change as flow changes in a nonlinear problem, it is possible that a smaller flow change than that determined in 5.4 will cause d_{k_E} to go to zero. If this were to happen, arc k_E would no longer violate complementary slackness. This section derives the value of flow change in arc k_E which drives d_{k_E} to zero.

An alternative form of d_{k_E} is useful here. The value d_{k_E} is the marginal cost change with respect to a flow change in arc k_E. Since the total cost changes only with the flow changes on basic arcs occasioned by the flow change on arc k_E, the value of d_{k_E} can be written:

Eq. (19):

$$d_{k_E} = \sum_{k(i,j) \in M_B} h'_k g_k + h'_{k_E}$$

Note that g_k is the marginal flow change in arc k with respect to a flow change in arc k_E and h'_k is the marginal cost. Since

$h'_k = h_k + 2Q_k F_N$, separating the linear and nonlinear terms yields:

Eq. (20):

$$d_{k_E} = \sum_{k \in M_B} h_k g_k + h_{k_E} g_{k_E} + \sum_{k \in M_N} (2Q_k F_N) g_k$$

This sum is over all the arcs in the basis since the value g_k is zero except for arcss on paths which are effected by a flow change in arc k_E. In the linear network, d_{k_E} is not effected as flow changes in arc k_E since h'_k is independent of flow. However, if a nonlinear arc k(i,j) has $g_k \neq 0$, changing flow effects h'_k and hence d_{k_E}. What one would like to be able to do is to compute the value of flow change in the entering arc at which d_{k_E} goes to zero.

Define:

Eq. (21):

$$d'_{k_E} = d_{k_E} + \Delta d$$

where d_{k_E} was the value before the flow change.

Again separating the linear and nonlinear terms:

Eq. (22):

$$d'_{k_E} = \sum_{k \in M_B} h_k g_k + h_{k_E} g_{k_E} + \sum_{k \in M_N} (2Q_k F'_N) g_k$$

In this equation, F'_N is the vector of flows in the nonlinear arcs modified as flow in arc k_E is changed.

Let:

Eq. (23):

$$F'_N = F_N + \Delta F_N$$

where F_N is the original flow and ΔF_N is the incremental flow in the nonlinear arcs.

Eq. (24):

$$d'_{k_E} = \sum_{k \in M_B} h_k g_k + h'_{k_E} g_{k_E} + \sum_{k \in M_N} (2Q_k F_N) g_k$$

$$+ \sum_{k \in M_N} (2Q_k \Delta F_N) g_k$$

The first three terms in this expression comprise d_{k_E} and the last term is Δd.

Solving for the flow change that makes $d'_{k_E} = 0$:

Eq. (25):

$$\Delta d = -d_{k_E}$$

$$-d_{k_E} = \sum_{k \in M_N} (2Q_k \Delta F_N) g_k$$

Define for the nonlinear arcs the vector G_N where:

Eq. (26):

$$G_N = \begin{bmatrix} g_1 \\ g_2 \\ \cdot \\ \cdot \\ \cdot \\ g_k \end{bmatrix} \quad \text{for } k \in M_N$$

Since Q is symmetric, equation 25 can be rewritten:

Eq. (27):

$$-d_{k_E}=2G_N^T\ Q\ F_N$$

From the previous discussion:

Eq. (28):

$$\Delta F_N=G_N f_{k_E}$$

Substituting for F_N in (27) yields:

Eq. (29):

$$-d_{k_E}=2G_N^T\ Q\ G_N f_{k_E}$$

Solving for the flow change that causes d_{k_E}' to be zero:

Eq. (30):

$$\Delta f_{k_E}=-d_{k_E}/2G_N^T\ Q\ G_N$$

where Δf_{k_E} is the change in flow f_{k_E}.

Note that since d_{k_E} is negative and Q is positive definite, the value of Δf_{k_E} will be positive. For the case where d_{k_E} is positive, a negative value for f_{k_E} results. This is appropriate since it indicates that the flow on arc k_E must be reduced to drive d_{k_E} to zero.

The result of the above formulation is a value for flow change in the nonlinear arc k_E which causes d_{k_E} to be zero. This limiting value is compared with the flow change determined by equation (16) or (17). The minimum of these two values will determine the amount of flow change to apply to arc k_E. If the limiting arc is determined by equation (16) or (17) arc k_E will enter the basis and the limiting arc will leave. If the limit is obtained by equation (30), the flow will be changed, but the basis will remain unchanged. In this case, the nonbasic nonlinear arc will have flow between its bounds.

The above represents a general development for solving the nonlinear nonseparable quadratic network.

This presentation justifies the inclusion of these nonlinear quadratic arcs in the network in their nonseparable form. To implement this new theory into the linear codes of Jensen and Barnes (1980), six new subroutines were created and several existing subroutines required some modification. These new codes were assembled in a package called NONLING and tested in the presence of positive and negative gains. These subroutines are flow charted in Part III of the Appendix.

CHAPTER VI

6. Estimating the Benefit Function

6.1 General

The preceding sections have shown that by exploiting the special structure of networks in a dynamic programming approach an expression can be derived which represents the future expected value of stored water. This was done by utilizing a single period network representation of a water resources system. Through the selection of discretized initial levels and the incorporation of stochastic inflows to these initial levels, the total water available to the period was determined. Subsequently, through successive random draws from the distribution of the stochastic inflows, enough data was assumed to be generated to allow a reasonably good approximation of a benefit function using a least squares fit of the observed network optimal solutions.

The water resources problem solution procedure requires two types of information in order to produce meaningful and useful results. The first type relates to the specific network parameters. These include all of the arc parameters, the river and reservoir data, demands, runoff distribution, etc. Naturally, this information is problem specific and would most generally be supplied by the user. A second class of information, however, is both model specific and user oriented and relates to the accuracy and credibility of the model. This class requires the answer to several questions.

1. How many discretizations of reservoir contents should be used and from the total number of possible level combinations generated, which should be used for an optimal design for this experiment?

2. What method of least squares regression should be used to fit the data obtained?

3. How many replications from the assumed known inflow distribution should be used for each level combination?

4. How can convexity be assured?

After addressing these four questions, and having observed some of the results, additional concerns require attention. These include:

5. Examining the variance covariance matrix to measure to validity of the least squares regression

6. Consideration of weighted least squares

7. Problems associated with weighted least squares

Finally, based upon all of the above analysis:

8. Determination of the data requirements for estimating the benefit function.

These questions and related issues will be addressed in the next 8 sections.

6.2 Design of the Experiment

The process of estimating the benefit function requires the discretization of the water levels in the system reservoirs. Consider a three reservoir system where each reservoir has a capacity of 25 units of water. Each unit will represent some specified number of acre feet. The hypothetical reservoir system of Figure 4-1 will be used throughout this discussion. This hypothetical example is further exercised in detail in Chapter 7. Note, there is no requirement for these reservoirs to be equal in size and the reservoir sizes are input parameters to the model, set by the user as dictated by his specific system.

Reservoirs are typically divided into zones according to their functional purpose. Linsley and Franzini (1964) define these zones as shown in Figure 6-1. For a very low level of water there is a minimum volume referred to as the dead storage pool. This is the level below the sluiceway where water cannot be released from the reservoir for other uses. At the very top of the reservoir, a flood pool is normally reserved. The purpose of this pool is to allow the temporary storage of water during heavy inflow conditions, thus preventing high water flood conditions downstream. During normal operations, This flood pool will usually be empty. Between these two pools is the useful storage volume. It is this portion of thereservoir that is of primary interest to this study. This useful storage is the portion of the reservoir that will be discretized for the dynamic programming approach. The percent of reservoir volume in the dead storage pool and in the flood pool is

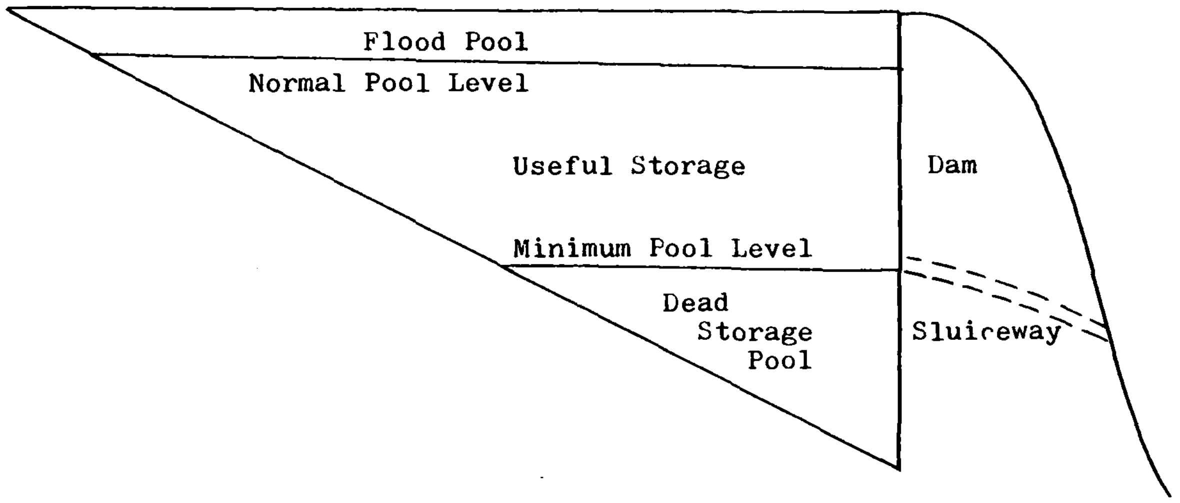


Figure 6-1

Reservoir Storage Pools

iependent upon the specific reservoir. Within this useful storage range, members of the Texas Water Development Board indicate that a reservoir holding only 50% capacity could be considered as having a severe shortage (depending upon the size and use of the reservoir). Thus, although the total useful storage pool is available, the user may be concerned about only a portion of it. Thus, the range to be discretized will be left to the user and he has complete flexibility (within the usable storage pool), in determining this range since the maximum and minimum water levels are input parameters to the model as a percent of reservoir capacity. Regardless of the range it is necessary to determine the number of levels or discretizations to use to produce a meaningful set of data, and finally, what subset of the total set of level combinations can be used to achieve an optimal design?

The significance of the number of discretizations lies in its effect on the accuracy of the quadratic fit and on the computational time required. Ideally, one should use as few discretizations as possible which yield the best "or near best" set of data for a quadratic fit. The total number of possible level combinations is equal to the number of levels (NN) raised to the number of reservoirs (R) power. Notationally, the number of level combinations is:

$$NN^R$$

regardless of the range.

For a three reservoir problem with three levels selected for each, there are 3^3 = 27 combinations of initial reservoir levels considered. For five levels there are 5^3 = 125 total combinations. Extending this to a larger problem, say six reservoirs and five levels there would be 15625 combinations.

As noted in the dynamic programming algorithm of Chapter 4, random inflows are added to the reservoir levels at the beginning of the period to yield total water available. Network solutions to these problems yield the desired observations which are used to fit a quadratic. Naturally, one does not need 15625 data points to fit a quadratic having 28 coefficients (which is the case for a six reservoir problem).

Assume for now the three reservoir case and three levels of discretization for each reservoir. The problem then is to select from the total set of level combinations an optimal set of design points. Using this terminology, a design point relates to a level combination Figure 6-2 shows a three level cubic representation for the three reservoir problem. Note, there are 27 distinct points on and within this cube represented by the intersection of lines (ignore the highlighted points for now). Each of these points represents a level combination or a design point. The goal is to select some subset from these 27 design points which will yield an optimal design. Box and Draper (1971) and Mitchel (1974a, 1974b) have defined an optimal set of design points for first and second order models. Their work is specifically for least squares approximations where the underlying error has a normal

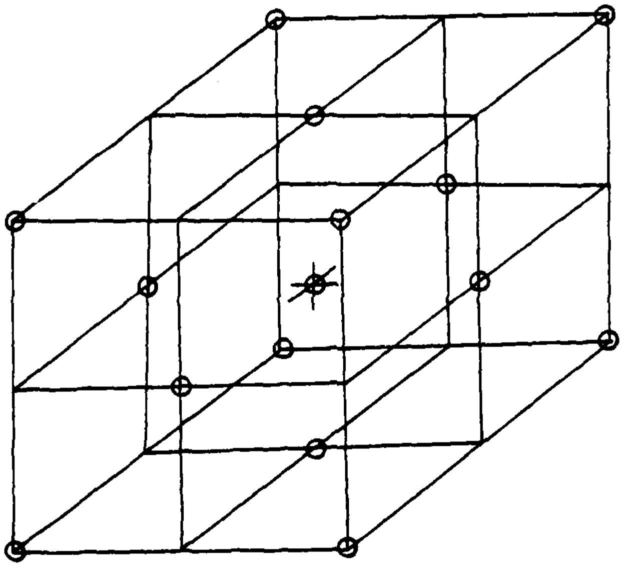


Figure 6-2

Three Reservoir, Three Level

Cubic Representation

distribution.

Two of the basic approaches to choosing an n-point experimental design are (i) to set down a simple factorial or fractional factorial design in the factors being studied, or (ii) to choose a design based on the well known det(X'X) criterion. Box and Draper (1971) indicate that the first method is much more simplistic and that the second is their preferred approach.

Details of their preferred approach are presented below and have been utilized in the computer program.

Before presenting their approach, a few new terms need to be discussed. A design point has already been defined as being equivalent to a level combination. A design point has dimension (1xR) for an R reservoir problem. The design matrix was defined in Chapter 4 as being the matrix of independent variables to be used for the regression. A full quadratic has been selected as the model to be fit by the regression. For the three reservoir, three discretization problem, this yields a design matrix of dimension (27x10) if all design points are used. To use the notation of Box and Draper and to be consistent with standard notation for regression analysis, this design matrix will be referred to as the X matrix.

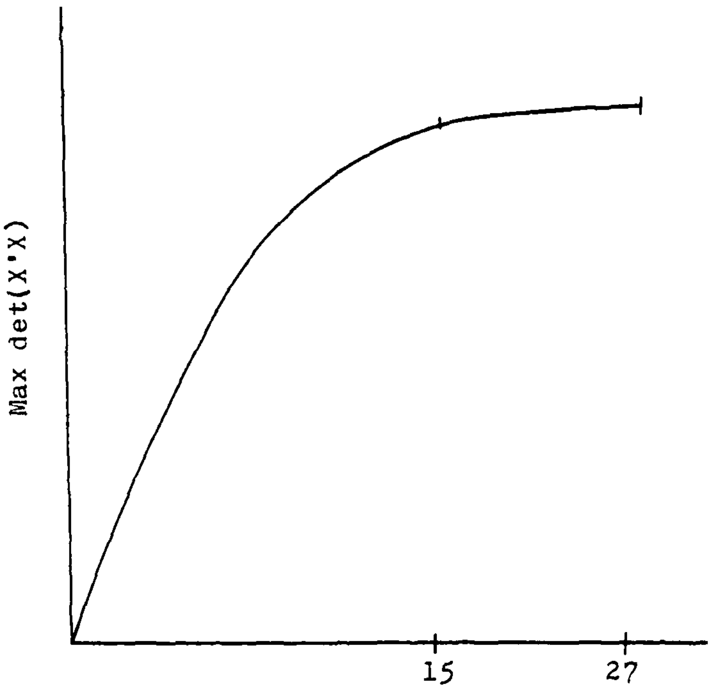
In their preferred approach, the criterion for designing experiments is based on maximizing the determinant of (X'X) indicated here as det(X'X), (X' is the transpose of the X matrix). Use of this criterion dates back to Smith (1918). The criterion has many appealing properties and its use has been justified in a number of ways. Box and Lucas (1959) indicated that its use leads to a confidence region for the parameter estimates of smallest hypervolume in parameter space. Kiefer (1961) showed that a design which maximizes det(X'X) also minimizes the maximum variance of any predicted value (obtained by using the regression function) over the experimental space. Further properties of this criterion are that it minimizes the generalized variance of the parameter estimates, and that the design obtained is invariant to changes of scale of the parameters. This is an important property not shared, for example, by the criterion: minimize trace $(X'X)^{-1}$, (i.e., min the average variance of.the parameter estimates).

Box and Draper (1971) considered several discretization techniques and found that the best design for a quadratic fit was to use three levels consisting of both end points and the midpoint of the selected range. These design points correspond to the 27 points of Figure 6-2. Thus, there are 27 design points from which to choose.

The choice of three levels of discretization runs contrary to the typical dynamic programming approach. Klemes and Doran (1977) specify 5-10 levels in their divided interval technique, while others say as many as 30 levels may be needed. However, these levels are not being used in the typical dynamic programming manner. The requirement is to select enough levels to allow a good approximation of a quadratic benefit function and three points are sufficient to fit a quadratic. In practice, once an expression which reflects the future value of water has been derived, the user can apply the actual water levels to this function resulting in an operating policy representative of the current situation.

What must be done next is to select n of these design points and apply the det(X'X) criterion, (n in this case can vary from 10 to 27). The design matrix for the full quadratic would now have dimension (nx10). For any given n, there are C^{27}_{n} subsets to consider, and for each n, there will exists a "best" design set in terms of maximizing det(X'X). For example, if n = 20 there are C^{27}_{20} = 388,030 different subsets of 20 design points to evaluate. Calculating the det(X'X) for all 388,030 of these subsets will result in one of them having a determinant that is greater than or equal to all others. The subset with the maximum det(X'X) represents the best set of design points to use, given that 20 are to be used. Now, plotting the optimal det(X'X) obtained for each n yields a form similar to the one shown in Figure 6-3. Through this process, Box and Draper were able to derive an optimal design for a quadratic which has a general structure and can be applied to problems of three or more factors.

The resulting optimal design is referred to as the "cube plus star" design. Mitchel (1974a, 1974b) refers to it as "D optimal" experimental design (the D referring to the determinant of the X'X matrix). This "cube plus star" nomenclature essentially describes the design. In Figure 6-2, the "cube" refers to the



Number of Design Points

Figure 6-3

Plot of Maximum det(X'X)

eight vertices of the cube. The "star" is defined as the center point on each of the six surfaces of the cube and one point exactly in the center of the cube. This results in a total of 15 design points of a possible 27 points in the three factor case. These 15 points are the highlighted points of Figure 6-2.

In general, if there are R reservoirs or dimensions to the problem, there will be:

$$2R + 2^R + 1$$

design points. This means that for the six reservoir case mentioned before, rather than using five discretizations for a total of 15625 design points, only three discrete levels and 77 total design points are required.

One possible disadvantage of the det(X'X) criterion is that it is a "variance criterion" and effectively assumes that the model considered is the true model. Box and Draper (1971) point out however that in situations where the design is physically restricted to a cuboidal region of interest, the difference between the spread of the design points for the best all-bias design and for the best all-variance design is minimal. Thus, they conclude, that the det(X'X) criterion appears to be not unrealistic either when the model is correct or when the design is restricted to the region of interest, or both.

The det(X'X) criterion applies to normal least squares problems where the variance about each design point is constant and there is no covariance present. Thus, the variance covariance matrix has the standard form:

$$\begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix}$$

The variance covariance matrix for the three reservoir problem will have dimension (15x15) where each row represents one of the 15 design points. The columns have the same meaning. The values on the main diagonal (σ^2) represent the variance about the mean of the calculated response for each design point. This variance is determined by taking a number of replications at each design point and calculating the variance using standard statistical methods. This variance is considered to be the same for all design points. This is the definition for homoscedasticity of data, Clark and Schkade (1974). All variance covariance matrices are symmetrical. This particular one has all off diagonal elements equal to zero. This indicates that there is no covariance present. This is necessary for the use of normal least squares regression which will be presented next.

6.3 Normal Least Squares

In the analysis of variance, for the normal least squares method:

$$b = (X'X)^{-1}X'Y$$

where:

X is the design matrix (independent variables) X'=X transpose

Y is the vector of dependent variables (mean of the replications)

b is the vector of estimates of the coefficients

The variance of the coefficients is found by:

Eq (1):

$$V(b) = (X'X)^{-1}(X'VX)(X'X)^{-1}$$

where V is the variance covariance matrix. This reduces to:

$$V(b) = (X'X)^{-1}\sigma^2$$

when σ^2 is known and V is considered to be the Identity matrix. For independent observations, this assumes that the variance covariance matrix of the design matrix is of the form:

$$V = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix}$$

Here as before, the diagonal element (i,i) represents the variance of the replications about the mean of the ith design point. For the water resources problem, σ^2 is not known. Consequently, an estimate of σ^2 must be derived by performing enough replications at each design point to get a reliable estimate of σ^2, which in this case is synonymous with the V matrix. Choosing the number of replications is the subject of the next section. Given that V has been derived of the form shown above, the variances of the estimated coefficients can be calculated using equation 1.

6.4 Replications

A means for determining the number of draws for deriving a good estimate of σ^2 is to select some acceptable relative error (alpha) which represents the ratio of the standard error of the estimate of the variance to the true variance.

It is known that:

$$\frac{(k-1)s^2}{\sigma^2} \equiv \chi^2(k-1) \Rightarrow \mathrm{Var}(\chi^2,(\Upsilon)) = 2\Upsilon$$

where k is the number of draws, χ^2 is the notation for the chi square distribution and Υ is the degrees of freedom for the chi square distribution.

$$\frac{(k-1)^2}{\sigma_i^4} \ \mathrm{Var}(s_i^2) = 2(k-1)$$

$$\mathrm{Var}(s_i^2) = (2\,\sigma_i^4) \,/\, (k-1)$$

$$\mathrm{SE}(s_i^2) = \sqrt{\frac{2}{k-1}}\ \sigma_i^2 \leq \text{alpha}$$

SE is the standard error of the estimate of the variance, s_i^2. Thus the relative error of the estimate to the true error is:

$$\frac{SE(s^2)}{\sigma_i^2} = \sqrt{\frac{2}{k-1}}$$

For fixed values of alpha k can be determined. A few values are shown in Table 6-1.

Table 6-1

Number of Draws Required to Meet Specified Accuracies

alpha	k
.30	22
.25	30
.20	100
.10	200
.07	500

If the process described by the reservoir system were relatively stable or unchanging from period to period, V could be estimated to the desired accuracy one time using a large number of draws. Then through the use of ordinary least squares equations, the variance of the coefficients of the benefit function for each period could be determined. For this problem, many things change from period to period, specifically, the inflows and arc parameters (including the Q matrix) for the nonlinear arcs. These changes require that the V matrix be reevaluated for each period since each period is in effect a new problem. This would suggest using as few draws as possible to keep the model computationally feasible. 30 draws will be used for the example problems and this is considered adequate for the water resources problem. If the user desires greater accuracy, naturally he can increase the number of draws at the expense of time.

6.5 Maintaining Concavity of Benefit Functions

A requirement of the optimization procedure is that each benefit function derived by the quadratic fit and subsequently used in the dynamic programming procedure must be a concave function. Thus for each t, BF(t) must be concave. This requirement comes from a limitation of the network flow optimization algorithm of Chapter 5. Specifically the algorithm only works to find a flow solution which minimizes total cost if the total cost function is convex. Since the negative of the benefit function is part of the cost function, this requires that the benefit function be concave.

This limitation of nonlinear minimization algorithms to convex objective functions is not uncommon. The presence of concave portions of the objective function may result in local minimums. Algorithms to handle the more general problem are usually much more complex and require more computation time for obtaining a solution. At any rate the network flow algorithm can handle only convex cost functions (or concave benefit functions).

The requirement of concave benefit functons does not seem to impose serious practical limitations for the water resources problem. It is clear that the marginal value of water stored for the future should be declining with the amount of water stored. This can be proved to be true for the models used for this research. However it is quite possible to happen upon a non-concave quadratic fit if proper precautions are not taken.

Even if the underlying model has a concave benefit, if independent random observations are taken at each design point statistical error may result in a convex fit. This is especially likely if the model is nearly linear. One thing that could be done is to check the quadratic matrix for negative definiteness after each benefit function is derived. However, in the event that the benefit function is not concave, an alteration of the fit would be required in order to continue. It is not clear how to perform this alteration in the general case. Instead of resorting to this manipulative alteration, a random sampling procedure will be used which will insure a concave form.

As indicated before, a series of random numbers which are used to derive the stochastic inflows is selected. If a different random number is selected for every design point, there is no assurance that the final fit will be a concave quadratic, regardless of the number of replications.

To assure concavity the same set of random numbers will be applied to every design point. Thus, if there are to be k replications for each design point, there will be a total of k random numbers. These same k random numbers will be used to generate the k replications for all levels.

To illustrate this idea, refer back to Figure 4-4. In this figure, there are L_R design points. The response matrix is $(L_R xk)$ where k is the number of draws. Note that the w^{th} draw is applied to all i levels of the design. Next, the Y_{iw} are averaged over the k replications and these mean responses are then used to fit the quadratic. Thus, it will be shown that by using the mean response values, obtained through the application of a constant set of random numbers, concavity will be assured.

Consider a particular design point i defined by the reservoir contents $S_{i(t-1)}$. A random draw w determines an inflow vector I_w. Since the inflows are assumed to occur at the reservoirs, the total water inputs to the system for design point i and random draw w is:

$$S_{i(t-1)} + I_w$$

These inflows appear in the network model as positive external flows for the reservoir nodes. The optimum flows are determined for the network model and the minimum cost is the response $Y_{i,w}$. The network problem has a convex objective function since the problem is linear except for the convex function -BF(t).

Let $Y_i(I_w)$ be the value of the minimum cost solution of the network model as a function of the vector I_w. It is well known that the minimum value of a convex objective expressed as a function of the right hand sides of the constraints is also convex. Thus, $Y_i(I_w)$ is a convex function with respect to I_w. The procedure used to obtain BF(t-1), samples from the distribution of I_w to obtain k distinct values. The k response values of $Y_{i,w}$ thus obtained must fall on the function $Y_i(I_w)$. By following the above procedure for the water resources problem, the possibility of generating response data that was nonconvex was removed.

6.6 Covariance Matrix Derivation and Analysis

Because of the requirement to use a constant random number set for all design points, a high degree of correlation between the design points has been induced into the model. The fact that this correlation is present requires the consideration of using a weighted least squares regression analysis.

Whether weighted least squares or ordinary least squares is used, it is required that a very good estimate of the variance covariance matrix (V) be derived. This is necessary since the variance associated with the problem is not known apriori. Because of the possibility of correlation being present it is necessary to derive this V matrix for use with both the ordinary and weighted least squares analysis.

The set up for weighted least squares analysis when all design points have common random inputs goes as follows:

Let:

$Y_{i,w}$ = response at the i^{th} design point when the random input is the w^{th} random sample from a given distribution

X_i = $(x_{i1}, x_{i2}, \ldots, x_{iu})$ = u-vector of settings of the independent variables at the i^{th} design point, $1 \leq i \leq n$.

where n is the number of design points used in the n-point optimal design.

The assumed model where B is the assumed coefficient is:

$$Y_{i,w} = \sum_{j=1}^{u} B_j x_{ij} + \mathcal{E}_{i,w}$$

where:

$$\mathcal{E}_{i,w} \sim N(0, \sigma_i^2), \quad 1 \leq i \leq n$$

and where σ_i^2 depends on i.

Now by supplying the same random input w to all n design points, the result is:

$$\begin{aligned} \mathrm{Cov}(Y_{i,w}, Y_{j,w}) &= \mathrm{Cov}(\mathcal{E}_{i,w}, \mathcal{E}_{j,w}) \\ &= V^{o}_{ij} \neq 0 \end{aligned}$$

With the notation:

$$V_o = (V^{o}_{ij}),$$

$$Y_w = \begin{bmatrix} Y_{1,w} \\ \cdot \\ \cdot \\ \cdot \\ Y_{n,w} \end{bmatrix}, \qquad B = \begin{bmatrix} B_1 \\ \cdot \\ \cdot \\ \cdot \\ B_u \end{bmatrix}$$

$$X = \begin{bmatrix} X_1 \\ \cdot \\ \cdot \\ \cdot \\ X_n \end{bmatrix} \qquad \mathcal{E}_w = \begin{bmatrix} \mathcal{E}_{1,w} \\ \cdot \\ \cdot \\ \cdot \\ \mathcal{E}_{n,w} \end{bmatrix}$$

the overall model in this situation becomes:

Eq(2):

$$Y_w = XB + \mathcal{E}_w$$

$$\mathcal{E}_w \sim N(0, V_o)$$

Averaging over K independent replications of the random input, yields:

Eq(3):

$$\bar{Y} = XB + \bar{\mathcal{E}}$$

where:

$$\bar{Y} = \begin{bmatrix} \frac{1}{K}\sum_{w=1}^{K} Y_{1,w} \\ \cdot \\ \cdot \\ \cdot \\ \frac{1}{K}\sum_{w=1}^{K} Y_{n,w} \end{bmatrix} = \begin{bmatrix} \bar{Y}_1 \\ \cdot \\ \cdot \\ \cdot \\ \bar{Y}_n \end{bmatrix}$$

$$\bar{\mathcal{E}} = \begin{bmatrix} \frac{1}{K}\sum_{w=1}^{K} \mathcal{E}_{1,w} \\ \cdot \\ \cdot \\ \cdot \\ \frac{1}{K}\sum_{w=1}^{K} \mathcal{E}_{n,w} \end{bmatrix} = \begin{bmatrix} \bar{\mathcal{E}}_1 \\ \cdot \\ \cdot \\ \cdot \\ \bar{\mathcal{E}}_n \end{bmatrix}$$

To apply weighted least squares (WLS) analysis to (3), the variance covariance matrix V for $\bar{\mathcal{E}}$ is required. Pick two design points i and j; compute:

$$\mathrm{Cov}(\bar{\mathcal{E}}_i, \bar{\mathcal{E}}_j) = \mathrm{Cov}(\frac{1}{K}\sum_{w=1}^{K} \mathcal{E}_{i,w}, \frac{1}{K}\sum_{m=1}^{K} \mathcal{E}_{j,m})$$

$$= \frac{1}{K^2} \sum_{w=1}^{K} \sum_{m=1}^{K} \mathrm{Cov}(\varepsilon_{i,w}, \varepsilon_{j,m})$$

Now $w \neq m \Rightarrow w$ and m are independent $\Rightarrow$

$$\mathrm{Cov}(\varepsilon_{i,w}, \varepsilon_{j,m}) = E(\varepsilon_{i,w}\, \varepsilon_{j,m}) - E(\varepsilon_{i,w})\, E(\varepsilon_{j,m})$$

The last two terms here are equal to zero by the assumption of the model.

This leaves:

$$E(\varepsilon_{i,w})\, E(\varepsilon_{j,m}) = 0$$

since w and m are independent.

On the other hand, for w = m:

$$\mathrm{Cov}(\varepsilon_{i,w}, \varepsilon_{j,m}) = \mathrm{Cov}(\varepsilon_{i,w}, \varepsilon_{j,w})$$
$$= V^{o}_{ij}$$

$$\mathrm{Cov}(\bar{\varepsilon}_{i}, \bar{\varepsilon}_{j}) = \frac{1}{K^2} \sum_{w=1}^{K} V^{o}_{ij}$$
$$= \frac{1}{K^2}\; K \;\; V^{o}_{ij} = \frac{1}{K} V^{o}_{ij}$$

$$V = (\mathrm{Cov}(\bar{\varepsilon}_{i}, \bar{\varepsilon}_{j})) = \frac{1}{K} V_{o}$$
$$= \frac{1}{K}(\mathrm{Cov}(\varepsilon_{i}, \varepsilon_{j}))$$

$$= \frac{1}{K}(\mathrm{Cov}(Y_{i,w}, Y_{j,w})$$

Thus to estimate V, compute:

Eq (4):

$$V_{ij} = \frac{1}{K} \mathrm{Cov}(Y_{i}, Y_{j}) =$$

$$\frac{1}{K} \left\{ \frac{1}{(K-1)} \left[\sum_{w=1}^{K} Y_{i,w}\, Y_{j,w} - K\bar{Y}_{i}\bar{Y}_{j} \right] \right\}$$

An example of a variance covariance matrix derived in this manner is shown in Table 6-2. Three things should be noted when observing this V matrix. (i) The off diagonal elements of this matrix are definitely not zero. This means that there truely is a significant covariance relationship as expected. This will be illustrated under (iii) below. (ii) For this V matrix, the main diagonal elements, which represent the variance of the replications about the mean response, are not significantly different. These diagonal elements represent the variances of the response about the mean of the various design points.

The test used to determine whether or not these variances are statistically equivalent is the Burr Foster $\tilde{Q}$ test statistic (Burr (1974)). $\tilde{Q}$ is used here to distinguish this test statistic from the Q (quadratic) matrix.

In terms of the sample variances s_i^2 computed within each treatment, the Burr Foster test statistic is given by:

$$\tilde{Q} = (\sum_{i=1}^{n} s_i^4) / (\sum_{i=1}^{n} s_i^2)^2$$

where n is the number of samples or design points in the optimal design. Large values of $\tilde{Q}$ lead to a rejection of the hypothesis of equal variances.

The calculated $\tilde{Q}$ statistic for the data of Table 6-2 is .0771. From a Burr Foster table of critical values, a $\tilde{Q}$ statistic

Table 6-2 Typical Variance Covariance Matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	62														
2	45	33													
3	45	33	33												
4	37	27	27	22				Symmetric							
5	50	37	36	30	40										
6	37	27	27	22	30	22									
7	34	25	25	20	28	20	19								
8	23	17	17	14	19	14	13	9							
9	43	32	31	26	35	26	24	16	30						
10	45	33	32	27	36	27	25	17	31	32					
11	46	34	34	28	37	28	26	17	32	34	35				
12	47	35	34	28	38	28	26	17	33	34	36	36			
13	35	26	25	21	28	21	19	13	24	25	26	27	20		
14	39	29	28	23	31	23	21	14	27	28	29	30	22	24	
15	40	29	29	24	32	24	22	15	28	29	30	31	23	25	26

of .08 or greater is considered significant at the .001 level with 29 degrees of freedom. Thus, the design points for the optimal design can be interpreted as having equal variances. (iii) The third feature to note when examining the V matrix of Table 6-2 is the correlation coefficient, denoted as P. This coefficient is calculated by:

$$P_{xy} = \frac{\text{covariance}(X,Y)}{(V(X)\ V(Y))^{1/2}}$$

It can be shown that $-1 \leq P_{xy} \leq 1$. The quantity P_{xy} is a measure of the association between the random variables X and Y. For example, if $P_{xy} = 1$, X and Y are perfectly correlated and the possible values of X and Y all lie on a straight line with a positive slope in the (X,Y) plane. If $P_{xy} = 0$ the variables are said to be unassociated, that is, linearly unassociated with each other. Calculating the value of P_{xy} in this manner for the data of Table 6-2, results in finding that all of the correlation coefficients are ≅ .99. It is quite clear from this data that X and Y are nearly perfectly correlated for this example. It is also quite clear that the off diagonal elements are not zero. This indicates that weighted least squares regression analysis should be considered.

6.7 Weighted Least Squares Regression Analysis

Derivation of the V matrix was done in section 6.6 using a set up for weighted least squares analysis. Estimation of the benefit function coefficients and their variance for weighted least squares is as shown below.

For the weighted least squares method:

$$b = (X'V^{-1}X)^{-1}X'V^{-1}Y$$

where V is the same variance covariance matrix as before.

If the observations were independent:

$$V = \begin{bmatrix} \sigma_i^2 & 0 & 0 \\ 0 & \sigma_i^2 & 0 \\ 0 & 0 & \sigma_i^2 \end{bmatrix}$$

where some of the σ_i^2 may be equal.

Recall however that taking the same sequence of random draws for each design point results in a highly dependent structure. Consequently the full variance covariance matrix (V) of the observations must be considered. This V matrix will be (15 x 15) where the i,jth entry will be:

$$\mathrm{Cov}(Y_{iw}, Y_{jw})$$

and the covariance is defined as:

Eq(5):

$$\mathrm{cov}(Y_{iw}, Y_{jw}) = \frac{1}{k}\left[\frac{1}{k-1}\left(\sum_{w=1}^{k} Y_{iw}Y_{jw} - k\overline{Y}_i\overline{Y}_j\right)\right]$$

This is the same expression obtained before as equation 4.

An equivalent expression for this Covariance which circumvents some of the numerical roundoff errors associated with these calculations is:

Eq(6):

$$V = \mathrm{Cov}(Y_{iw},Y_{jw}) = (\frac{1}{k})\left[\frac{1}{k-1}\left(\sum_{w=1}^{k}(Y_{iw}-\overline{Y}_i)(Y_{jw}-\overline{Y}_j)\right)\right]$$

Given this V matrix calculated as equation (5) or (6), the variance of the coefficients for weighted least squares analysis is expressed as:

Eq (7):

$$V(b) = (X'V^{-1}X)^{-1}$$

6.3 Covariance Matrix Singularity

The fact that there is a highly correlated structure due to the procedure for selecting and applying common random draws for all design points cannot be ignored. This fact should suggest the use of a weighted least squares approach. Because of this significant correlation, a near singular V matrix is derived. For most problems this occurs for the period T data. In these cases use of the weighted least squares formulas, requiring inversion of the V matrix simply do not work.

The reason for this singularity is that due to the method of applying common random numbers, there is a strong dependency between responses over the entire design space. So strong in fact that the correlation coefficients for the covariance are nearly all

.99. This suggest that all of the responses lie in or very near a hyperplane in 15 space. For the case of 15 design points, given the other 14 points, the 15th can be predicted with very good accuracy. This interpretation follows from the fact that the noninvertability of the V matrix implies:

$$\det(V) \cong 0$$

This is the same as saying there exists a vector "a" such that:

$$a'Va = 0$$

which implies:

$$\mathrm{Var}\left(\sum_{i=1}^{15} a_i Y_i\right) = 0$$

This then comes from the definition of a hyperplane in 15 space:

$$\sum_{i=1}^{15} a_i Y_i = C$$

Now, a hyperplane in 15 space can be thought of as shown in Figure 6-4. Some of the problems start out as invertable weighted least squares problems, however, all of the responses fall within this hypervolume in 15 space. Initially, this hypervolume may be large enough or thick enough that the V matrix is invertable. Taking the result of the inversion and estimating the next set of coefficients has the effect of squeezing down this hypervolume until eventually it is too thin to allow an invertable form of the V matrix.

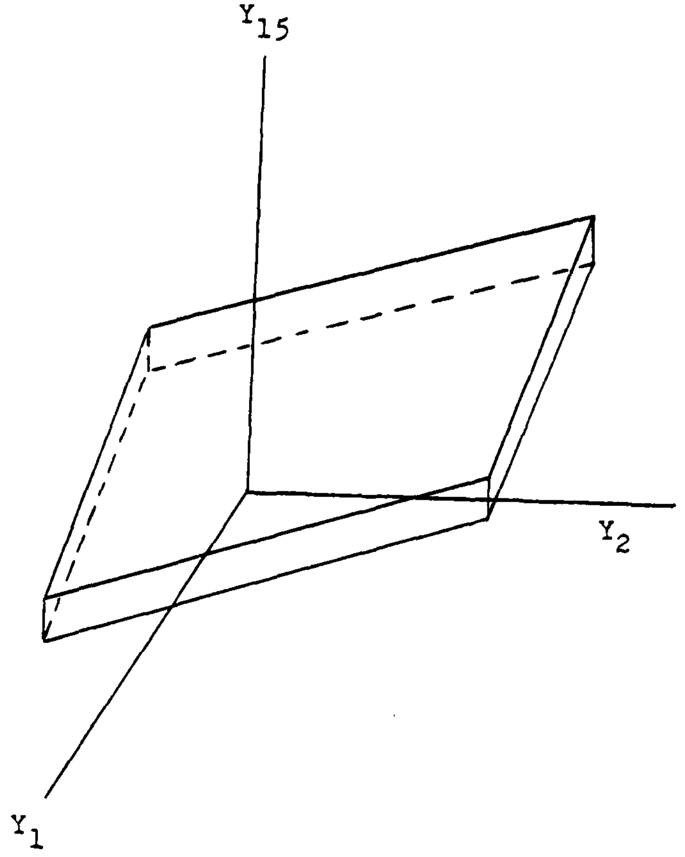


Figure 6-4

Illusion of Hypervolume in 15 Space

To support this theory a linear regression was performed on the response data using $\overline{Y}_1$ as the dependent variable and the remaining 14 mean responses as the independent variables. The model to be fit was thus:

$$\overline{Y}_1 = B_o + \sum_{j=2}^{15} B_j \overline{Y}_j + \epsilon$$

The results were even more convincing than anticipated. The results of this regression are shown in Table 6-3(a).

Next a similar regression was performed on a set of data that was originally invertable and observed that in fact this matrix was nearly singular from the outset. Table 6-3(b) shows the results of this regression.

These tables conclusively support the theory that the means of the responses all lie in a hypervolume in 15 space. Further, this totally explains the noninvertibility of the variance covariance matrix.

Thus, while weighted least squares analysis is indicated in nearly all of the data, its use brings about its own demise due to the noninvertibility of the V matrix. Consequently, the recommended methodology does not utilize estimation of the coefficients using weighted least squares.

Table 6-1 in section 6.3 showed the number of draws required to get some specified accuracy of estimating the V matrix. If the problem did not change from period to period, the V matrix could be estimated one time using a large number of draws, and then a fewer number of draws could be used for subsequent periods.

Table 6-3

Results of Linear Regression
(Y_1 with Y_2 through Y_{15})

	(a) Not Invertible	(b) Invertible
R-Squared	1.0	1.0
Standard Dev.	.00835	.1104
Residual(SS)	.00599	.18281
F-Value	354422760.	1400488.
Significance	.0	.000

Since the same number of draws will be required for each period another criteria for determining the number of draws is to select k' so as to control the maximum variance of the predicted error. This is different from the derivation of the number of draws for the V matrix in the following way. Recall that in deriving the V matrix equation (4) was used where:

Eq (8):

$$V_{ij} = \frac{1}{K} \mathrm{Cov}(Y_i, Y_j)$$

Here K is the number of draws which is determined by the selection of an acceptable relative error as in Table 6-1.

Controlling the maximum variance of the predicted error requires that the variance covariance terms derived in equation (4) be used but without the division by K. Let:

Eq (9):

$$V_o = KV$$

This is the desired V matrix for making this determination. Now for ordinary least squares it is desired to select k' large enough to control the maximum variance about the mean of the worst design point, or:

Eq (10):

$$\max_{x_o \in X} \left(x_o(X'X)^{-1}(X'V_oX)(X'X)^{-1}x_o'\right)^{1/2} (t_{a/2},df)/\sqrt{k'} \leq \mathrm{Del}$$

where

x_o = row in X matrix with largest variance

k = number of draws

t = student t value for desired accuracy with given d.f.

Del = absolute tolerable error (% of Y_o)

Rearranging:

Eq (11):

$$k' = \left(\frac{t_{a/2,df}}{Del}\right)^2 \max_{x_o \in X} (X_o(X'X)^{-1}(X'VX)(X'X)^{-1}X_o')$$

The results obtained using the normal least squares calculations as applied to the data of Table 6-2 follow. In this case, the initial V matrix was obtained using 30 draws. By applying equation (9) to this matrix and using the result (V_o) in equation (11), k' was determined to be approximately 20.

Table 6-4 shows other results given various settings of alpha and the tolerance level in absolute error.

Table 6-4

Number of Draws for Controlling Maximum Variance

Number draws for V	Tolerance level	alpha setting	Calculated # draws
30	.05	.01	20
30	.01	.04	30
100	.01	.05	33
100	.01	.01	82

From this table it would seem reasonable to use either 30 draws (alpha = .04) or approximately 90 draws (alpha = .01) depending upon the desired accuracy. These two choices seem reasonable since in both cases the number of draws used to estimate V and the calculated number of draws are nearly equivalent. As was done in section 6.3, 30 draws will be used. Again, the user can use more draws if greater accuracy is desired.

6.9 Selected Methodology

Given the statistical information just presented, it is desired to select a prudent and statistically sound heuristic for the general problem. Both the accuracy of the model and its computational efficiency must be considered. Experience has shown that each three reservoir nonlinear network requires approximately .0115 seconds of CPU time. Due to the dynamic nature of the water resources problem, it is necessary that the V matrix be updated every period. Accordingly, the selection of the number of draws (K) to estimate the V matrix and the number of draws (k') for the periods to follow should ideally be the same.

For these reasons, it is felt that a minimum of 30 draws should be used for estimating the V matrix and for subsequent use in calculating the variance of the estimated parameters. The estimation of the parameters will be calculated using ordinary least squares methods, and if the variances of the parameters is calculated, it will be done using the full V matrix in the manner of equation 4.

While the main interest lies in the estimate of the parameters it is instructive to know what levels of variability exists. By using 30 draws per level for all periods a reasonably good estimate of the parameters will be achieved with a minimum of computational time.

The relative error of the standard error of the estimated variance to the true variance is 25%. Given this 25% criteria, very high confidence about this estimate is realized as indicated by the 1% absolute tolerance level and the 4% alpha limit from the t table (see Table 6-4).

In summary, it is felt that 30 draws is a good compromise between accuracy and computational time. For 30 draws per level and 15 levels per period the computational time will be roughly 7 seconds per period for the three reservoir problem. The number of periods to use is highly dependent upon the specific problem being considered, its intended use and the availability of data. This will be addressed more fully in Chapter 7.

CHAPTER VII

7. Example Applications

This chapter includes several example applications to demonstrate the feasibility of the model. Section 7.1 includes two network formulations with some variations applied to both. Section 7.2 includes a representative application to the Guadalupe River Basin in Texas by attempting to evaluate a proposed new system to meet the demands of the year 2020.

7.1 Hypothetical Problems

This section includes both a three reservoir and a four reservoir example problem. There are two versions of the three reservoir problem, the differences being in the number of arcs and the selection of the arc parameters.

Figures 7-1, 7-2 and 7-3 represent the three problems to be considered. These will be referred to herein as examples 1, 2 and 3 respectively. Although the node structure of Figures 7-1 and 7-2 are the same, note the differences in the number of arcs and in the arc parameters. The network of Figure 7-2 assigns a benefit to having a minimum amount of water in the river reaches. This may be necessary due to hydroelectric concerns or perhaps due to concerns for aquatic life. Additionally, the benefits for supplying demand

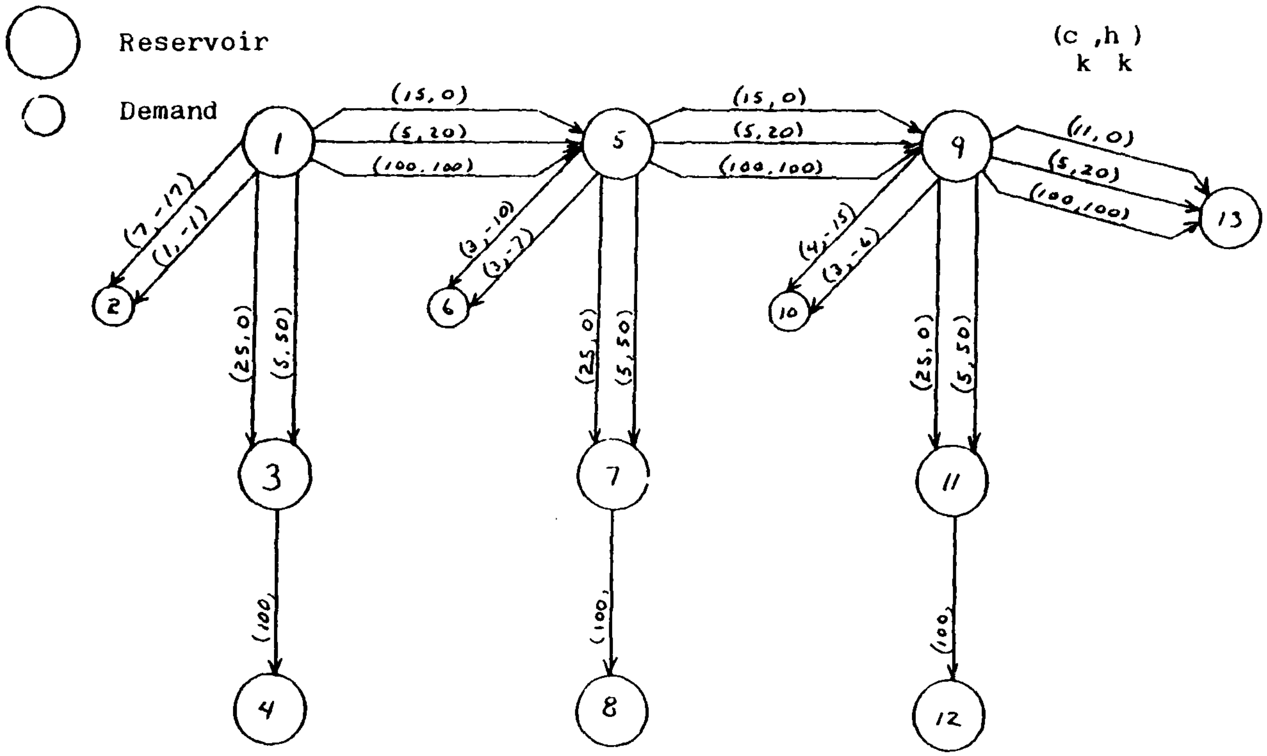


Figure 7-1

Three Reservoir (Example 1 Model)

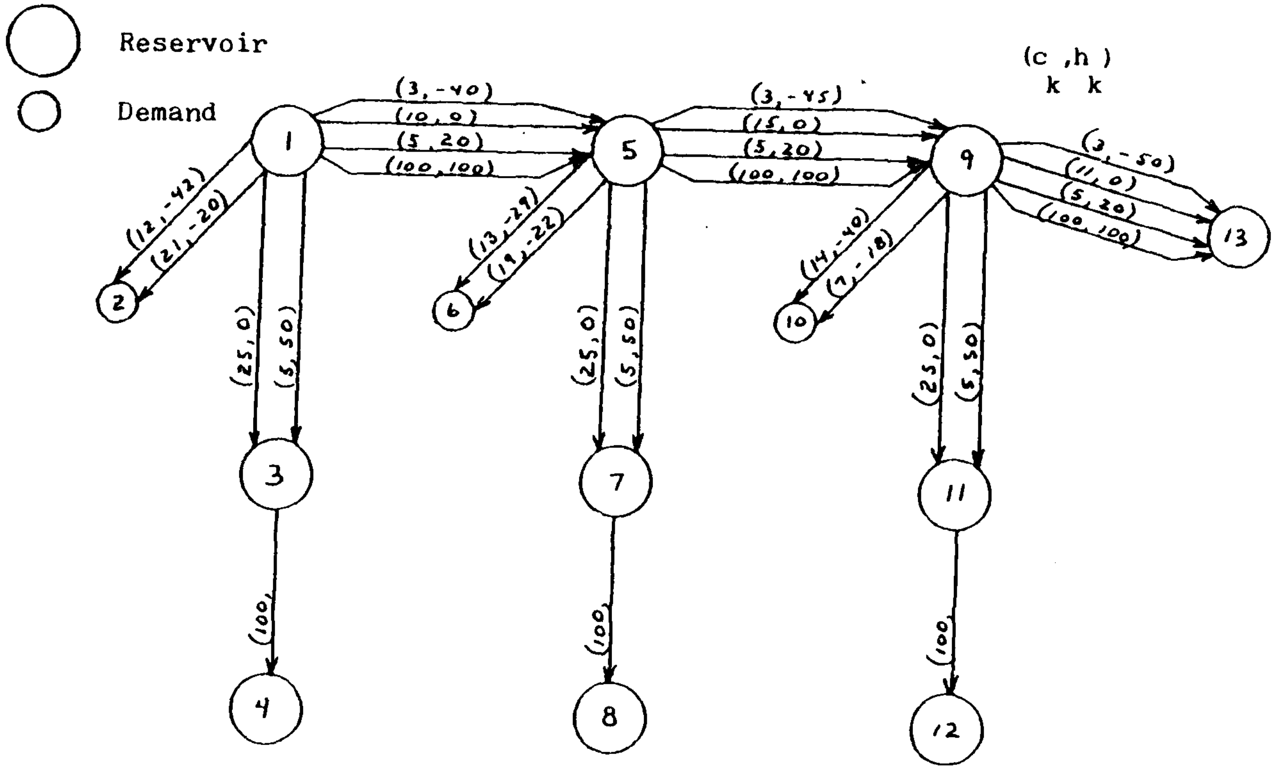


Figure 7-2

Three Reservoir (Example 2 Model)

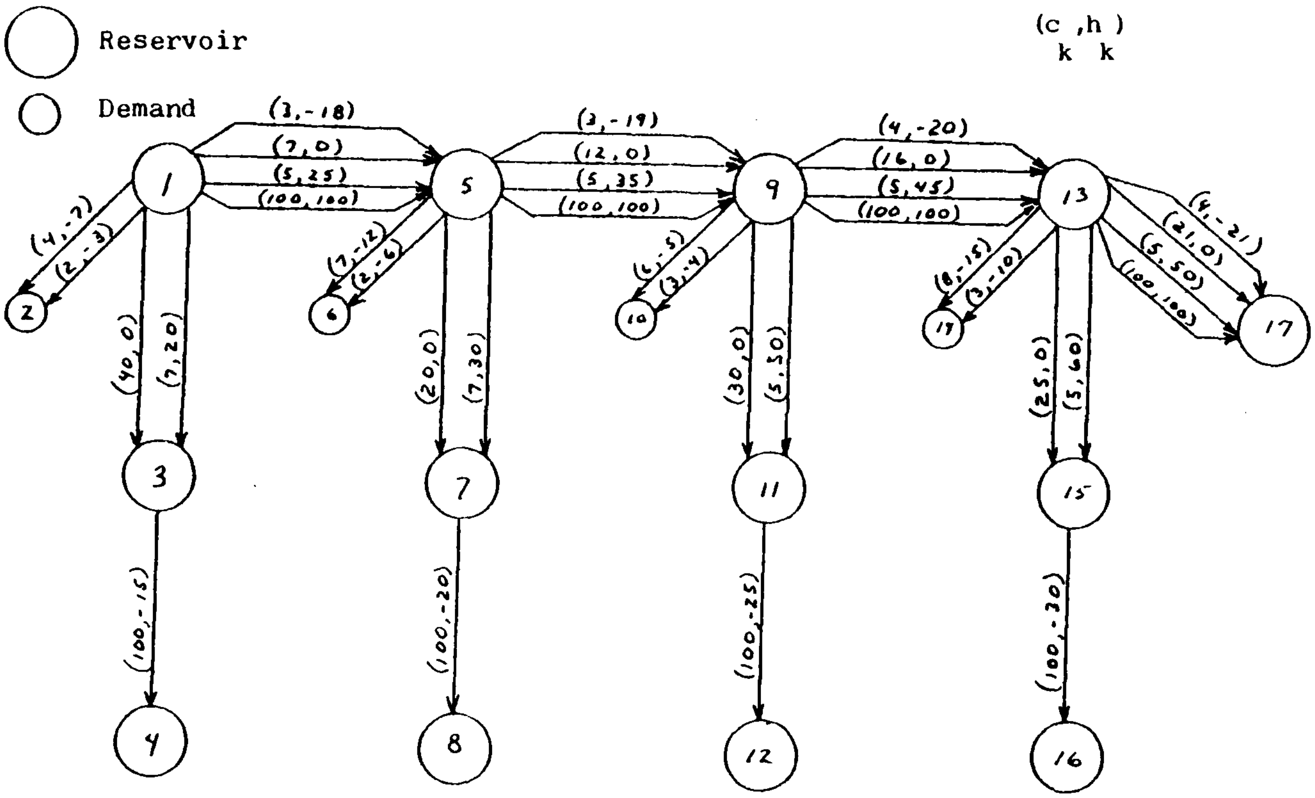


Figure 7-3

Four Reservoir (Example 3 Model)

are revised in this figure which tend to place more value on meeting the current needs than the benefits for demand of Figure 7-1.

Reference has been made throughout this report to Figure 4-1. The network of Figure 7-1 is identical to Figure 4-1 and many of the experimental results are used throughout this report.

For Figures 7-1 and 7-2, all reservoirs have a capacity of 25 units of water with an allowance for 5 additional units as a costly high water condition. The discretizations are (.5, .7, .9). For Figure 7-3, the capacities of the reservoirs are (40, 20, 30, 25) for nodes (1,5,9,13) respectively. For this example the same discretizations of (.5, .7, .9) are used. In this chapter, Q (the quadratic matrix) will use the subscript t. This will mean that if T = 12, Q_T will be the assumed quadratic benefit function for period T, and the dynamic programming algorithm will generate Q_t for t=11, 10, 9,...,1 in this order. Additionally, when Q_t is used, it will imply the existance of the associated linear terms of the full quadratic, in addition to the quadratic matrix itself. Using this notation, Q_T for Figures 7-1 and 7-2 is taken to be the negative of the benefit function used as the example in Chapter 5. Q_T for the example of Figure 7-3 uses the identity matrix for the quadratic matrix. Q_T terms for these problems are summarized in Table 7-1 which includes the quadratic terms and the associated linear terms. For both cases, the Q_T terms were arbitrarily selected.

Table 7-1

Assumed Conditions for Q_T

		Figs 7-1,7-2	Fig 7-3	
f_1	B1	-59.22	-15.0	f_1
f_2	B2	-46.61	-20.0	f_2
f_3	B3	-39.69	-25.0	f_3
f_1^2	B4	.86	-30.0	f_4
f_2^2	B5	.53	1.0	f_1^2
f_3^2	B6	.52	1.0	f_2^2
f_1f_2	B7	.64	1.0	f_3^2
f_1f_3	B8	.40	1.0	f_4^2
f_2f_3	B9	.68	0.0	f_1f_2
	B10	--	0.0	f_1f_3
	B11	--	0.0	f_1f_4
	B12	--	0.0	f_2f_3
	B13	--	0.0	f_2f_4
	B14	--	0.0	f_3f_4

In analyzing these problems it is interesting to observe the effects of two primary factors, (i) the nature of the inflow data and (ii), the effect in the long run of the assumed values for Q_T. These two issues will be addressed in the next two sections.

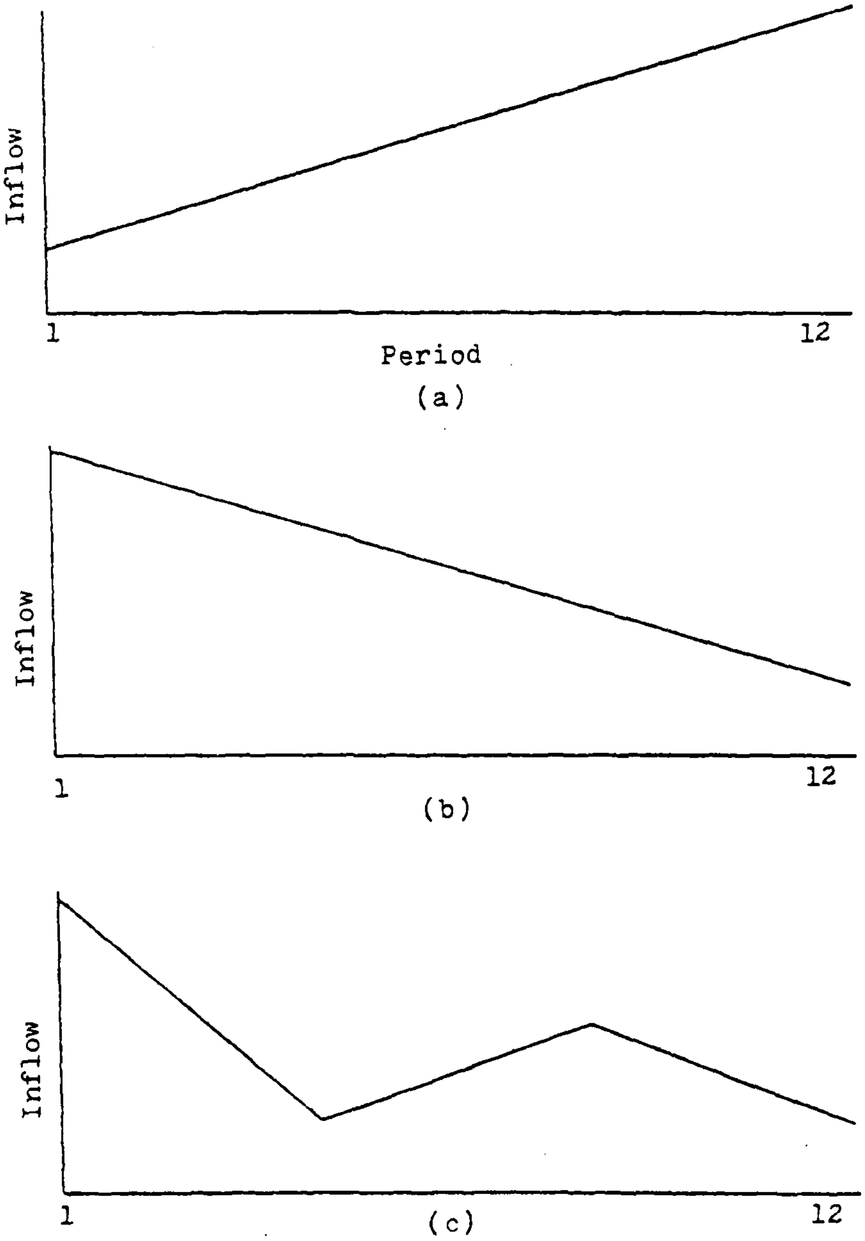
7.1.1 Model Response To Inflow Data

To address the first issue Q_T was arbitrarily selected as shown in Table 7-1 and the inflow pattern was varied. Three variations of the inflows were considered as shown in Figure 7-4a, b and c. Figure 7-4a represents inflows which are low in the present and monotonically increase to higher conditions in the future. Since the algorithm works backward from the future to the present it is expected that the current decisions will not insist that water be stored as a first priority since the model foresees more supply in the future.

Figure 7-4b represents the opposite inflow situation to that just discussed. Here there is more water in the present but less is anticipated in the months to come. It is expected in this case that the model will attempt to save some water for future use.

Finally, Figure 7-4c depicts the water fluctuating throughout the time horizon going from wet to dry then wet to dry again. This may be more representative of the inflow profile over a longer time horizon. For the examples, T was selected to be 12 with each period being equivalent to one month.

These inflow characteristics and the network of Figure 7-1



(a)

(b)

(c)

Figure 7-6

Inflow Profiles

will be referred to as example 1-1, 1-2 and 1-3. Similarily, applying these inflow profiles to examples two and three will yield three cases each.

For all of these problems, inflows were assumed to be distributed normally with standard deviations running from 50% to 30% of the mean for all reservoirs. Actual inflow data for the profiles of Figure 7-4 are included in Part I of the Appendix.

The negative of the derived benefit functions for Q_1 for each of the example 1 cases are shown in Table 7-2. Tables 7-3 and 7-4 show similar results for examples 2 and 3 respectively.

Some comments regarding the data of Tables 7-2,3,4 follow. There are two primary things to note when evaluating these tables.

The first item of interest for these tables deals with the magnitude of the linear terms of the benefit function when compared to the other negative costs of their respective networks. As an example, consider Table 7-4 and Figure 7-3. Consider the routing of flows as the flows into the system at the reservoir nodes are increased from an initial zero level. The first unit of flow will be routed based solely on the linear cost contribution, since for the nonlinear arcs, there is no quadratic contribution to the marginal cost at zero flow. Consequently, for example problem 3-1 which foresees a wet future, the initial allocation of water is to meet current demands. This is seen by comparing the negative benefit for supplying demand with the negative linear terms in the benefit function. Users 6 and 14 will first be supplied water

Table 7-2

Negative of Benefit Functions for Example 1 ($t = 1$)

		Ex 1-1	Ex 1-2	Ex 1-3
Constant	B0	-89.7	-249.2	-375.9
Linear	B1	-17.2	-25.9	-67.3
	B2	-11.8	-15.9	-60.2
	B3	-11.7	-15.7	-38.9
Squared	B4	.004	.274	1.29
	B5	.053	.098	1.07
	B6	.057	.102	.67
Cross Product	B7	.002	.128	2.0
	B8	.002	.114	1.33
	B9	.068	.181	1.47

Table 7-3

Negative of Benefit Functions for Example 2 (t = 1)

		Ex 2-1	Ex 2-2	Ex 2-3
Constant	B0	-502.6	-815.9	-2071.8
Linear	B1	-42.3	-43.2	-49.9
	B2	-42.1	-43.6	-34.4
	B3	-42.1	-43.6	-38.8
Squared	B4	.006	.026	.33
	B5	.036	.082	.104
	B6	.036	.081	.34
Cross Product	B7	.0004	.005	.07
	B8	.0004	.005	.01
	B9	.071	.164	.09

Table 7-4 Negative of Benefit Functions for Example 3 (t = 1)

		Ex 3-1	Ex 3-2	Ex 3-3
Constant	B0	-438.7	-132.7	538.3
Linear	B1	-10.8	-71.8	-114.7
	B2	-10.2	-68.5	-73.6
	B3	-9.5	-60.3	-62.2
	B4	-9.6	-27.5	-33.1
Squared	B5	.036	2.39	2.12
	B6	.026	1.73	.96
	B7	.026	1.35	.83
	B8	.034	.56	.52
Cross Product	B9	.056	2.22	1.99
	B10	.039	1.09	1.15
	B11	.037	.40	.43
	B12	.044	1.47	1.25
	B13	.043	.58	.49
	B14	.047	.78	.51

until their minimum demands are met (saturation of one of the demand arcs). User 14 will also supply its secondary demand before any flow is allocated to the reservoir for future storage. Users 2 and 6 will compete for water almost immediately with their reservoirs.

For the inflow profile of Figure 7-4b, the future is expected to be dry. In this case, note the extreme change in the linear terms of the nonlinear arcs which indicate a strong desire to store the initial allocation of water, (likewise for the inflow profile of Figure 7-4c). This pattern also holds for the data of Table 7-2. Table 7-3 is unique in that the linear cost terms assigned to the nonlinear arcs do not change much at all over the three inflow profiles. This is a result of placing large negative costs on the river arcs of Figure 7-2. For this example, the current needs dominate the entire process and in the end, all demands (both users and storage) reflect near equal priorities.

The second item of interest is the amount of quadratic effect realized. For all three example problems, as the future supply of water decreases, the quadratic terms increase. Similarily, for inflow profile 1, these terms are very small indicating a near linear situation. Thus, as the future supply of water decreases, interactive forces tend to arise which supports the hypothesis of reservoir interaction.

One final comment on these tables. The data for inflow profiles 1 and 2 cannot be compared directly with profile 3 because there is more water overall in profile three. For profile 3, period 1 has a very large amount of inflow, as do the next few periods. Thus, the results for this profile are altered by current and near term high inflow conditions. This could be what is causing the biggest difference in the coefficients.

As an additional display of the model results, Table 7-5 shows one of the negative benefit functions along with the calculated standard deviations for the estimated coefficients. These standard deviations were calculated using the methods described in Chapter 6. These results are very similar for all example problems and indicate that in fact, 30 draws does yield very good estimates of the coefficients. An additional test was performed using this data. Rather than estimate the coefficients and calculating their variance using the normal equations, the data was provided as input to a least squares regression package (FIXREG). The disadvantage of using FIXREG regularily is that it requires considerably more time due to thenumerous statistics it generates and to its built in plotting capability. However, it was used on occasion to assure that meaningful results were being generated. An example of some of the results are also shown in Table 7-5. These t and F levels of significance and the overall R^2 term conclusively support the quadratic as a valid function for the regression. This example is highly representative of the results for all example problems.

Table 7-5

Statistical Results for Coefficients

	Q_1	Std.Dev	T-STAT
B0	-250.9	28.8	8.7
B1	-25.9	.38	68.2
B2	-16.0	.19	84.0
B3	-15.8	.19	83.0
B4	.274	.012	23.0
B5	.099	.002	49.5
B6	.103	.002	51.5
B7	.127	.007	18.1
B8	.114	.008	14.3
B9	.184	.004	46.0

R^2 = .999+

F Significance = 5.8E+5

AD-A106 769   AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH   F/G 13/2
A MODEL FOR SOLVING MULTIPERIOD MULTIRESERVOIR WATER RESOURCES --ETC(U)
MAY 81 D D COCHARD
UNCLASSIFIED   AFIT-CI-81-48D   NL

3 of 3
AD
A106769

END
DATE
FILMED
12 81
DTIC

7.1.2 Operational Use of Benefit Functions

Once the benefit functions have been derived by the dynamic programming procedure, they should be useful in an operational context to make decisions on the optimal distribution of available surface water in any given month. The function can easily be saved by storing the coefficients of the quadratic form and the linear coefficients. This use of the benefit functions is illustrated in this section using the benefit functions derived for period 1 for the example cases.

To use the benefit functions the decision maker must observe his reservoir levels at the beginning of the month, calculate or observe the expected inflows and run the single period optimization one time. The resulting flows will determine the policy for period 1.

To implement this activity with the computer codes used for the dynamic programming procedure a few changes need to be made to the data set. The required changes are very simple and do two things. First, they force the logic to skip certain calculations that pertain to multiperiod and mult-draw problems. Second, specific changes to the input data set allow the user to specify his desired conditions.

Using the Q_1 benefit functions of Tables 7-2, 3 and 4, several period 1 conditions were assumed. These assumed conditions represent the sum of the observed reservoir levels and the expected inflows for the period (ie. total water available for period 1).

With these total inflows the optimum flows were obtained by the network algorithm. These flows describe the optimum decision set for each condition. Some of these are shown in Table 7-6.

Specifically, the data of Table 7-6 represents two examples to be discussed. The first two columns of Table 7-6 reflect two inflow level sets for period 1 as applied to example problem 1-2. The third and fourth columns reflect similar inflow level sets for example 1-3. There are other, perhaps more interesting, examples which involve more releases and transfers of water between reservoirs, but these were selected to demonstrate the network optimization technique. In all these cases (except the last column) all the water available at each reservoir was used to supply demand at the reservoir or saved in the reservoir. No water was transferred or released.

This first discussion pertains to example 1-2 of Table 7-6. Consider the first column with inflows to the three reservoirs of 13/13/13. Referring to Table 7-2 and Figure 7-1, it is clear that the linear cost coefficients for the benefit function are more negative than the demand cost coefficients. For reservoir 1, this means that the initial allocation of water will go to the reservoir. The question is how much water will be stored before any demands are met? As water is stored in all reservoirs, the marginal cost on the nonlinear arcs for storing water increases. At some point, the marginal cost will be greater than the marginal cost for supplying demand. For reservoir 1 this occurs at 11.35

Table 7-6

Single Period Network Results for Given Total Water Availability

			Example 1-2		Example 1-3	
			Assumed Total Water		Assumed Total Water	
	Marginal Cost	Maximum Demand	13/13/13*	22/22/22	13/13/13	18/18/18
User 1	-17	7	1.65**	7.0	6.69	7.0
	-1	1	--	--	--	--
User 2	-10	3	--	3.0	--	3.0
	-7	3	--	.11	--	3.0
User 3	-15	4	4.0	4.0	4.0	4.0
	-6	3	--	--	3.0	3.0
Amount Stored						
Res 1			11.35***	15.0	6.31	11.0
Res 2			13.0	18.9	13.0	12.0
Res 3			9.0	18.0	6.0	5.027
Releases						5.972

* These numbers indicate the total water available at the three reservoirs

** These numbers indicate the amount allocated to the reservoirs

*** These numbers indicate the amount stored for tie future units of water. At this point, the marginal cost for storing water in reservoir 1 is increased to -17. This is the breakpoint for supplying demand, and the next 1.65 units are allocated to the demand at reservoir 1. If these 1.65 units were stored rather than used, the marginal cost would be greater than -17 and hence, the network flows would not be optimal.

With regards to reservoir 2 and demander 2, all 13 available units are stored. At this level, the marginal cost for reservoir 2 is -10.24. The marginal cost must increase to -10.0 before flow will be allocated to demand. Finally, for reservoir 3, the initial allocation of water went to the reservoir since -15.7 < -15.0. However, very quickly, the marginal cost for this reservoir drops to -15.0 and the next four units of water go to the user (node 10). Once user 3 received 4 units, it is again profitable to store water and the remaining water is in fact stored. Final marginal cost for reservoir 3 is -10.19, far better than supplying an additional 3 units to user 3 at a cost of -6.0 per unit.

For the 22/22/22 column of example 1-2, similar flows will occur for the first 13 units. However, with an additional 9 units of water available at each reservoir, user 1 receives all 7 units requested with the remaining 2.65 units being stored. In this case the ending marginal cost for reservoir 1 is -13.22 versus -17.0 from before. User 2 now receives all 3 units demanded at a cost of -10.0 and an additional .11 units at -7.0. Given additional flow, the second increment of demand at user 2 will be satisfied before any additional storage. At the current levels, the marginal cost for reservoir 2 is -7.0. Reservoir 3 ending marginal cost is less than -6.0 and user 3 does not receive its second increment of demand. Note, in all these cases, no flow is released downstream. This would not occur unless one of two things were to happen. First, if an upstream reservoir has an over abundance of water such that some marginal cost downstream (user or storage) was profitable, or 2, if a reservoir has enough water to cause its marginal cost to go to zero while meeting all demands. Then it would be profitable to release water at a cost of zero rather than store it at some positive cost (since additional storage would cause the marginal cost to go positive).

Turning now to the data for example 1-3, for inflows of 13/13/13 reservoir 1 marginal cost goes to -17.0 with only 6.31 units of flow with the next 6.69 units going to user 1 at a cost of -17.0. This breakpoint is far lower than for example 1-2 due to the significantly higher quadratic terms of Table 7-2. Reservoir 2 ending marginal cost is -10.8, still less than -10.0 for its user. And for reservoir 3, its ending marginal cost was -3.45 (nearing zero).

These seemingly low ending reservoir levels are apparently due to the high inflows in the first few periods of profile 3.

For the 18/18/18 inflows of example 1-3, the important thing to note is that ending marginal cost for reservoir 3 is zero.

In this case rather than store additional water (since all demands are met), 5.972 units are releasd at zero cost.

Many variations of these problems were run (not always having the same inflows for all reservoirs) with different results for each case.

Some interesting alternatives to these examples might be to allow transfer of water back upstream at zero or some very low cost thereby making it profitable to spend a few dollars to get the water where it is needed the most rather than release it at zero cost. Another interesting alternative would be to supply a large amount of water to reservoir 1 with zero inflows at reservoirs 2 and 3. If the zero cost arcs of the system were given a large capacity, enough water could be put into the system to meet all demands (maximize the current return) and to drive the marginal cost for all reservoirs to zero (maximize the future return). This could be done using slack inflow at reservoir 1 at zero cost and by putting a high penalty on releasing water to the ocean. In this case the objective function would achieve its absolute minimum, thus deriving the solution to the quadratic problem.

As indicated throughout this report, the reservoir contents at the beginning of the period do not have to correspond to the discretized reservoir water levels used to derive the benefit functions. The functional form of the benefit function was derived from the discretized levels, but this form is a continuous form and applies for any set of water level combinations that fall within the selected range of interest.

7.1.3 Effect of Q_T on the Model

The main concern here is the effect that the assumed values for Q_T have on the model. Naturally, all other arc cost parameters are also questionable since selecting them is an art in itself. However, it is assumed that the user will have a fairly good feel for these values. What he will not have a good feel for is the future value of water at the end of the time horizon.

To measure this effect, the network and inflows for example 1-2 were used. Several variations of the period T quadratic benefit functions and linear terms were examined with nearly identical results in all cases. The results shown in Table 7-7 represent three of these conditions. This table shows all 12 of the Q_t benefit functions as they were derived over time. For the three cases shown, case 1 started with the assumed benefit function of Chapter 5. Case 2 kept the same linear terms, but used the identity matrix as the quadratic matrix. Case 3 used the same quadratic matrix as case 1 but changed the linear terms. Observing the first few periods of results (periods 12,11,etc.), it is quite clear that they are significantly different. However, it is noted that after approximately 8 periods of data (Q_4) the coefficients of the benefit functions are very close in value. After 12 periods, all three Q_1 benefit functions are nearly identical. Many other variations were tried with similar results. Consequently, it is determined that for these problems, a minimum of 8-12 periods are

Table 7-7

Benefit Function Convergence

Period	Coeff.	Case 1	Case 2	Case 3
Q_T	B0	0.0	0.0	0.0
	B1	-59.22	-59.22	-50.22
	B2	-46.61	-46.61	-40.61
	B3	-39.69	-39.69	-30.69
	B4	.8633	1.0	1.0
	B5	.5320	1.0	1.0
	B6	.5226	1.0	1.0
	B7	.6340	0.0	0.0
	B8	.4066	0.0	0.0
	B9	.6770	0.0	0.0
Q_{11}	B0	-571.	-351.	-475.
	B1	-25.38	-49.11	-22.68
	B2	-25.41	-32.10	-21.37
	B3	-24.99	-31.25	-21.67
	B4	.155	.770	.147
	B5	.248	.430	.317
	B6	.313	.490	.437
	B7	.111	.011	.000
	B8	.076	.000	.000
	B9	.287	.263	.081
Q_{10}	B0	-203.	-316.	-179.
	B1	-19.72	-34.44	-18.93
	B2	-19.94	-27.00	-17.53
	B3	-22.46	-28.18	-17.53
	B4	.048	.420	.046
	B5	.154	.270	.181
	B6	.266	.350	.271
	B7	.031	.004	.000
	B8	.026	.001	.000
	B9	.208	.291	.098

Table 7-7 (continued)
page 2

Q9	B0	-171.	-276.	-153.
	B1	-18.26	-25.56	-17.90
	B2	-17.89	-23.71	-15.71
	B3	-20.15	-24.87	-17.83
	B4	.023	.211	.023
	B5	.116	.190	.121
	B6	.201	.253	.214
	B7	.012	.002	.000
	B8	.011	.001	.000
	B9	.184	.287	.108
Q8	B0	-134.	-172.	-123.
	B1	-17.68	-21.15	-17.67
	B2	-16.77	-22.52	-14.71
	B3	-18.14	-23.20	-16.68
	B4	.014	.101	.017
	B5	.091	.161	.083
	B6	.139	.189	.156
	B7	.005	.001	.000
	B8	.005	.001	.000
	B9	.165	.280	.112
Q7	B0	-208.	-260.	-197.
	B1	-17.52	-18.58	-17.25
	B2	-15.47	-20.06	-14.01
	B3	-16.36	-20.63	-15.35
	B4	.013	.041	.007
	B5	.079	.133	.073
	B6	.113	.157	.128
	B7	.003	.001	.001
	B8	.002	.001	.000
	B9	.141	.241	.104

Table 7-7 (continued)
page 3

Q_5				
	B0	-207.	-237.	-197.
	B1	-21.01	-21.19	-20.94
	B2	-14.96	-17.49	-14.24
	B3	-15.29	-17.75	-14.76
	B4	.114	.119	.114
	B5	.077	.106	.076
	B6	.096	.120	.105
	B7	.029	.028	.027
	B8	.022	.023	.020
	B9	.136	.198	.119
Q_5				
	B0	-204.	-226.	-199.
	B1	-24.48	-24.59	-24.39
	B2	-15.34	-16.37	-14.84
	B3	-15.53	-17.06	-15.16
	B4	.219	.220	.220
	B5	.081	.098	.082
	B6	.097	.111	.106
	B7	.063	.064	.061
	B8	.054	.056	.052
	B9	.147	.186	.136
Q_4				
	B0	-223.	-236.	-220.
	B1	-23.61	-23.73	-23.52
	B2	-14.65	-15.58	-14.32
	B3	-14.38	-15.79	-14.67
	B4	.208	.211	.210
	B5	.075	.084	.076
	B6	.088	.095	.096
	B7	.058	.057	.055
	B8	.052	.053	.049
	B9	.140	.163	.134

Table 7-7 (continued)
page 4

Q_3				
	B0	-176.	-186.	-174.
	B1	-25.88	-26.0	-25.86
	B2	-16.67	-17.14	-16.43
	B3	-16.52	-16.96	-16.34
	B4	.248	.252	.252
	B5	.101	.105	.103
	B6	.111	.113	.117
	B7	.124	.124	.122
	B8	.096	.097	.092
	B9	.179	.190	.176
Q_2				
	B0	-215.	-221.	-213.
	B1	-28.80	-28.83	-28.81
	B2	-18.79	-19.04	-18.58
	B3	-18.03	-18.28	-17.85
	B4	.322	.322	.331
	B5	.140	.142	.141
	B6	.141	.142	.149
	B7	.195	.195	.193
	B8	.146	.146	.144
	B9	.232	.238	.235
Q_1				
	B0	-249.	-250.	-247.
	B1	-25.92	-25.92	-25.92
	B2	-15.94	-16.00	-15.73
	B3	-15.74	-15.82	-15.57
	B4	.274	.274	.278
	B5	.098	.099	.098
	B6	.102	.103	.104
	B7	.128	.127	.126
	B8	.114	.114	.111
	B9	.181	.184	.178

required to dampen out the effect of the assumed benefit function for period T

7.1.4 Computation Times

This section includes a brief summary of computation times. Table 7-8 shows the CPU times and I/O times required for these problems. All problems were run on the University of Texas CDC 6600 system. The results indicate that the algorithm requires approximately .0115 seconds of CPU time for each three reservoir nonlinear network and approximately .028 seconds for each four reservoir network.

7.1.5 Benefit Contours

Before going on to the application to the Guadalupe River Basin, it is interesting to observe certain properties of the model as it relates to the hypothetical examples. Specifically, for the three reservoir problems, it is possible to plot iso-benefit curves as a function of two reservoirs while holding the third reservoir at some specified level. This may be important for some systems that for one reason or another require that a given reservoir be closely regulated and maintained at or near a precise level. Any of the three reservoirs can be specified as fixed for the algorithm.

As an example, suppose it is of interest to fix reservoir three at 20 units of water for example 1-2. In so doing it is

Table 7-8

Summary of CPU and I/O Times (seconds)

Problem	CPU Time	I/O Time
1-1	58.8	10.6
1-2	70.6	11.0
1-3	64.9	10.5
2-1	68.3	10.5
2-2	54.0	10.7
2-3	60.2	10.6
3-1	258.9	12.6
3-2	267.5	12.6
3-3	246.4	12.5

Network Solution Times

Number of Reservoirs	Number of Networks Solved	Average CPU Time	Time Per Network
3	5400	62.23	.0115
4	9000	257,6	.028

desired to observe the resulting iso-benefit curves.

A program (CONT1) has been written which takes the desired benefit function and target total benefits as input data. This program then calculates and plots the desired curves of equal benefit. Figure 7-5 shows a typical plot for this problem. In this figure it is observed that a major part of the ellipses lie outside the feasible region determined by the reservoir capacities. There are an infinite number of these ellipses, depending upon the target total benefit. For Figure 7-5, the center of the ellipse is far outside the feasible region. Recall that for this problem, all reservoirs were restricted to a maximum capacity of 25 units with 5 additional units allowed at a penalty. Thus, these curves are only meaningful in the region of 30 units or less for each reservoir. If the feasible region were unbounded, the center of the ellipse would represent the maximum future benefit given reservoir 3 (f_3) is held at 20 units. If f_3 were not fixed, the true optimal solution for this benefit function could be obtained using the following:

$$\frac{\partial f}{\partial x_i} = 0 \quad \text{for } i = 1,2,3$$

The results for this problem are $f_1 = 34.5$, $f_2 = 29.6$ and $f_3 = 32$ with a maximum benefit of -1140.

Now letting $f_3 = 32$, a similar plot of benefit functions is obtained as shown in Figure 7-6. This plot, as expected, has its center at (34.5, 29.6) with the desired benefit of -1140. This point indicates the maximum of the benefit function. The user

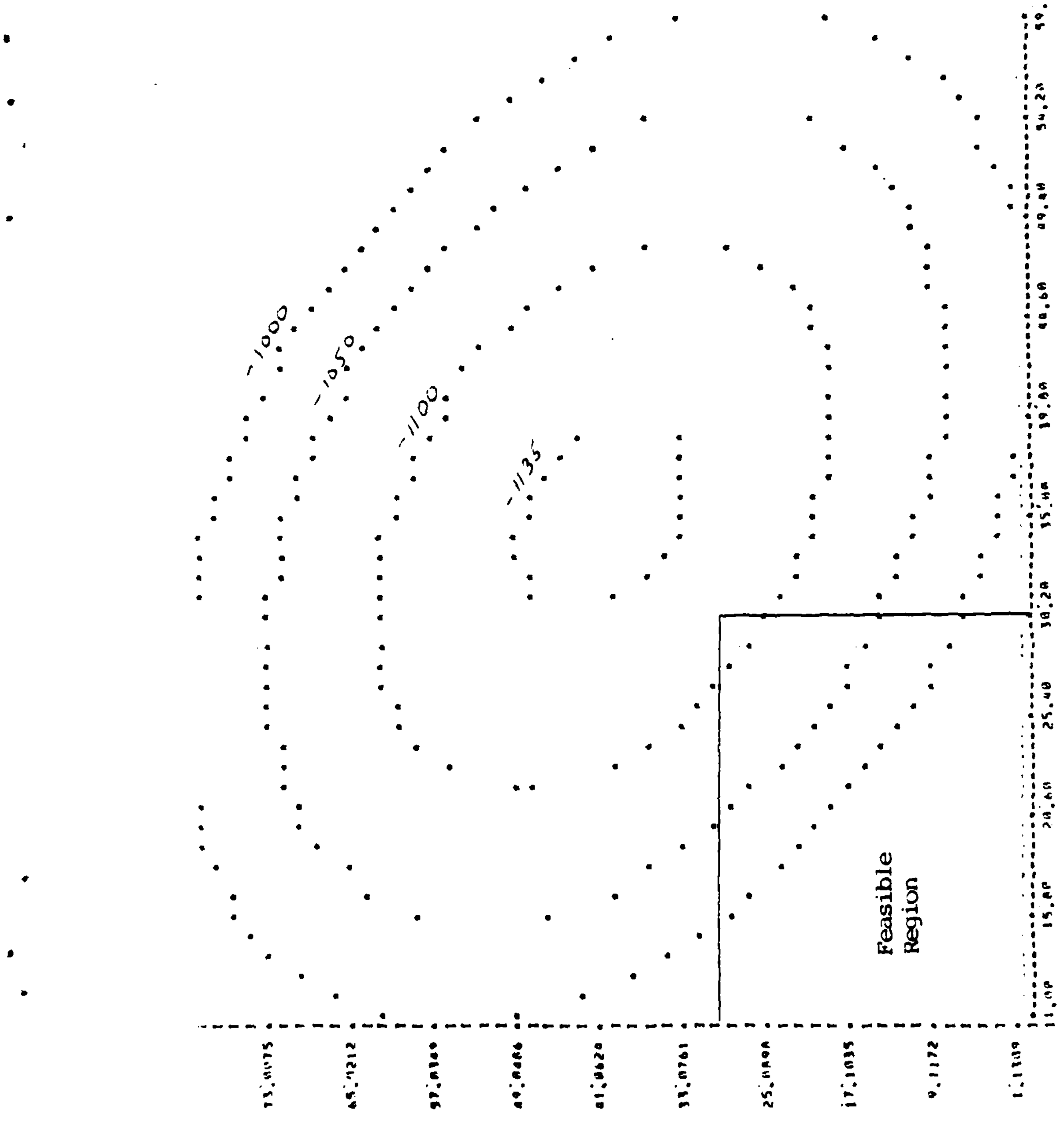


Figure 7-5 Full Range Ellipse

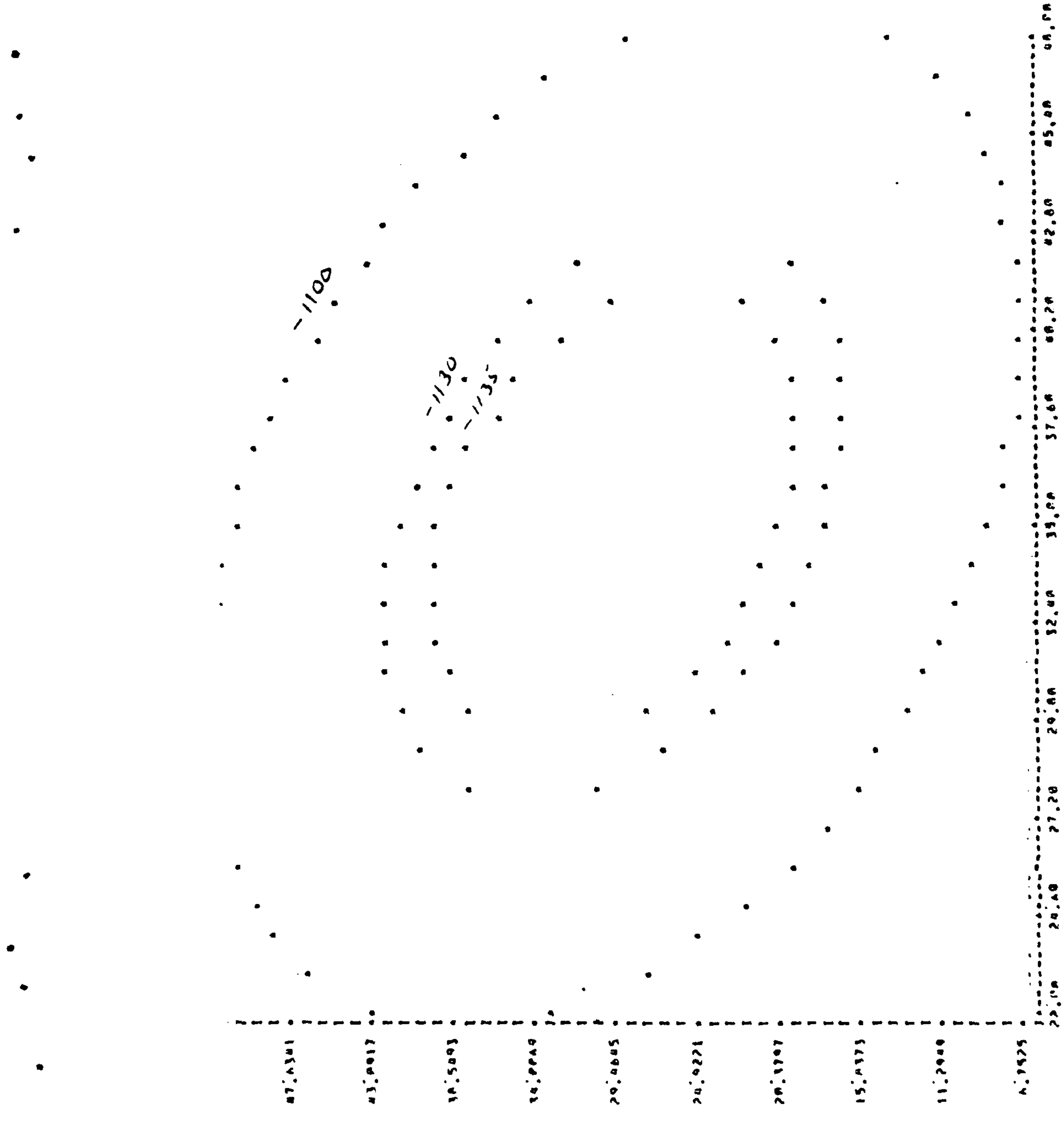


Figure 7-6 Optimum Ellipse, Future Only

would not operate here however since he must also consider the tradeoffs with the current returns. Also, the function is certainly not valid outside of the feasible region.

As mentioned before, some of these ellipses fall mostly outside the feasible region. It is somewhat more useful to concentrate on the feasible region only. Figure 7-7 has its range limited to a maximum value of 30. For the data of example 1-2, only parts of the ellipses are contained in this region, and the maximum benefit attained is at the upper right hand corner of the feasible region. In this case, $f_1 = 30$, $f_2 = 30$ and $f_3 = 20$. The value of the benefit function at this point is -1112. This is as expected since this example attempts to force the storage of water for the future. Given a fixed amount of water in the system, it is unlikely that these levels would be stored due to the tradeoff of current requirements and the fact that to reach levels of 30 in any of the reservoirs would require that penalty arcs be used. While this is allowed and could be profitable, it would depend on the costs assigned to other network arcs.

Another set of plots taken from a different trial problem (and hence a different benefit function) had the form of Figure 7-8. The center of the ellipses are contained in the feasible region. This type of curve will occur when the model determines that it is not too important to save water for the future, (example 1-1 data).

One interesting thing to note regarding these ellipses is

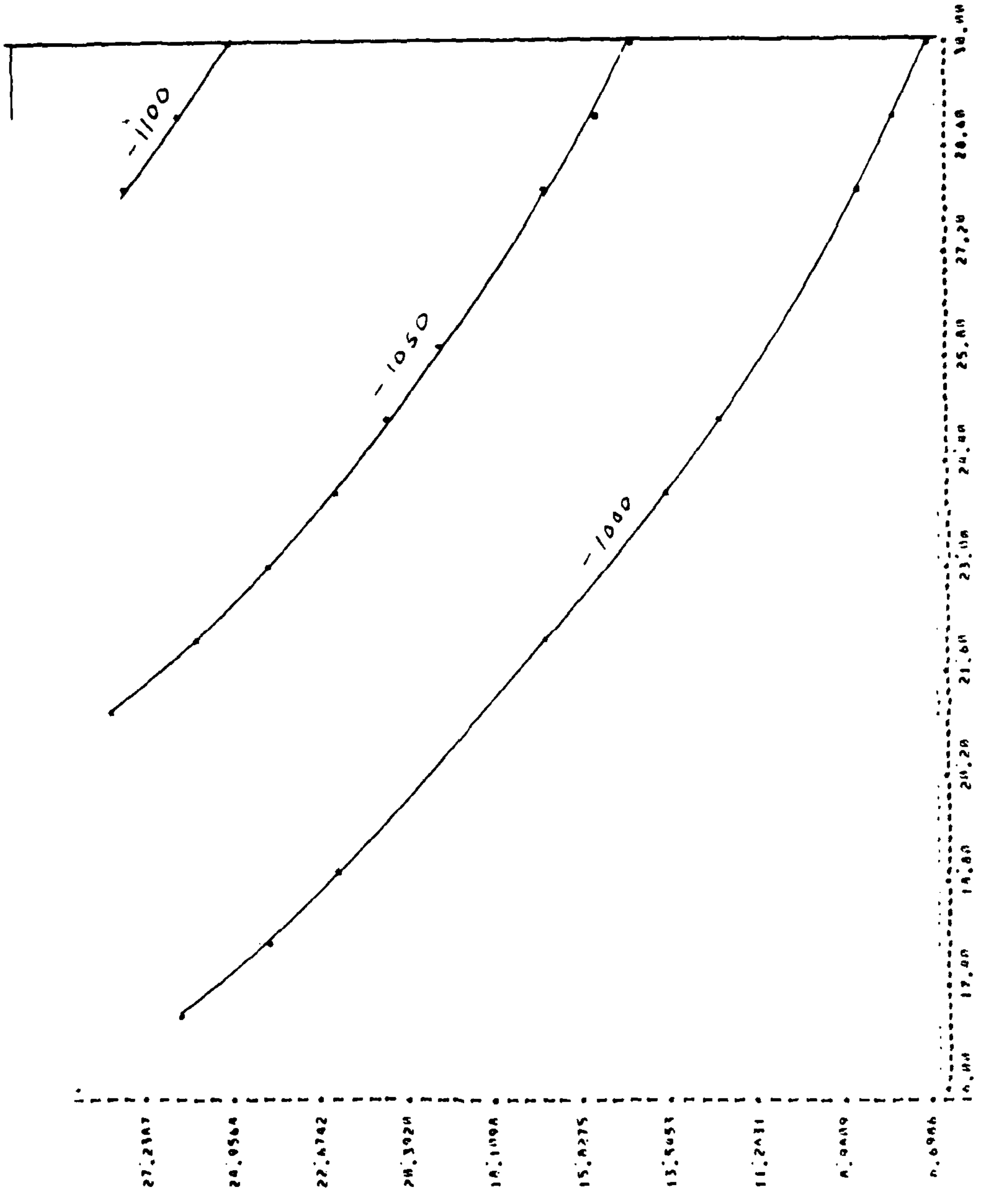


Figure 7-7 Feasible Region of Figure 7-5

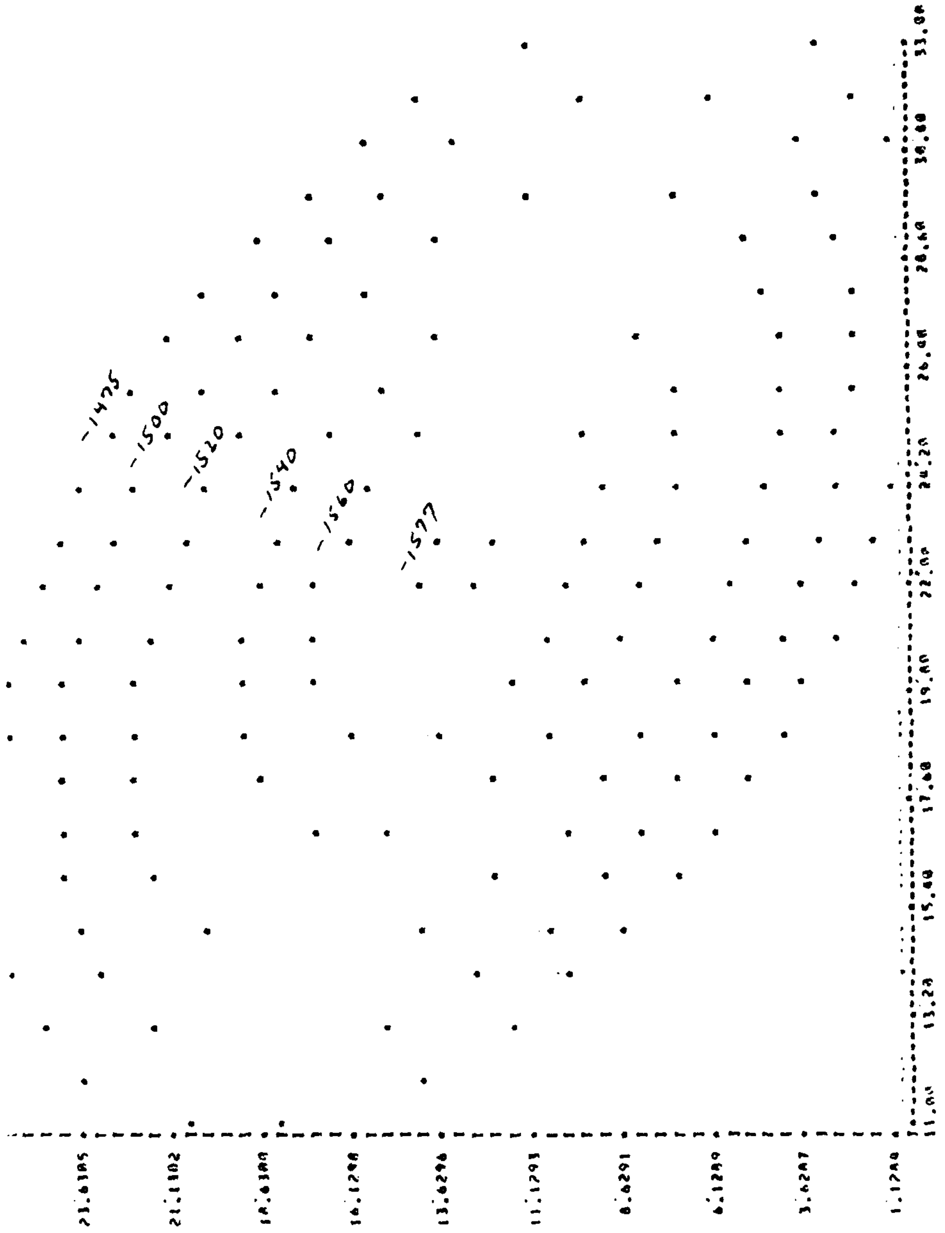


Figure 7-8 Full Range Ellipse Within Feasible Region

the amount of rotation that they have. This rotation is an indication that some interaction between the plotted reservoirs is present. If no interaction were present, the main axes of the ellipses would parallel the x and y axes.

7.2 Guadalupe River Basin - Stochastic Case

The Guadalupe and San Antonio River Basins are situated in the southern Coastal Bend area of Texas. From their headwaters in the Edwards Plateau region of Central Texas, the two rivers cross the Gulf Coastal Plains and combine into a single stream shortly before entering San Antonio Bay. The average annual rainfall over the two drainage basins varies from 35 inches near the Gulf Coast to 25 inches in the area of the headwaters.

The water needs in the basins are presently supplied largely from groundwater formations underlying the region, with the principle source of groundwater being the Edwards and associated Limestone Aquifer. This underground reservoir is a water supply source for irrigation and for many municipalities, including the city of San Antonio. The Edwards Aquifer is also the source of water for the Comal and San Marcos Springs. These springs provide the major portion of base flow in the Guadalupe River.

Canyon Reservoir is the only existing major storage reservoir in the Guadalupe River Basin. The reservoir provides both water supply and flood control storage. Six small hydro-electric dams on the Guadalupe River downstream from New Braunfels constitute the only other significant impoundments in the Guadalupe Basin.

The consumptive water requirements within the Guadalupe Basin are currently being adequately supplied from a combination of surface and subsurface sources. However, with the greatly increased water demands projected for this area, it is evident that judicious water resources management will be essential in order to fully supply future water needs.

The major demand center for these two basins is the San Antonio metropolitian area. The city and adjoining suburbs have experienced rapid population growth in the past several decades, and are projected to have greatly increased populations in the future. The area's water needs in the past have been supplied exclusively by pumpage from the Edwards Aquifer, however, this municipal and industrial pumpage added to the irrigation pumpage in the Balcones Escarpment area to the west of the city is presently approaching the average annual recharge into the Edwards formation. Should the increased future irrigation, municipal and industrial water requirements continue to be supplied from groundwater, the Edwards Aquifer will undergo drastic reductions in water levels, diminishing springflows and severe deterioration in water quality.

As an alternative to this depletion of the Edwards Aquifer, the Texas Water Development Board (1975) has proposed the development of a conjunctive ground and surface water resources system to supply the San Antonio area. The plan calls for limiting San Antonio pumpage to 215,000 acre-feet annually, and supplying the remaining water requirements with surface water conveyed through pipelines from a system of reservoirs in the San Antonio and Guadalupe River Basins. This application will evaluate the proposed changes to the Guadalupe River Basin.

A number of reservoir and pipeline projects were considered as possible components in the optimal water supply system. Of concern here is the addition of three reservoirs to go along with the existing Canyon Reservoir. These three new reservoirs include Cloptin Crossing, Cuero I and Cuero II. A schematic of the proposed sytem was shown in Figure 3-16.

The deterministic solution to the Guadalupe River Basin application was discussed in Chapter 3 and shown in the Appendix. The deterministic inflows were taken to be the mean values of the inflows by month over the 46 year period. The demands represented the projected incremental demands over the 1970 requirements. Total demands are not used since this system was designed to satisfy the demand growth, with the assumption that the current methods for supplying demand would continue to be used at their current levels. In this case it was shown that sufficient water would be available on the average to meet the projected demands for the year 2020. Based upon this result, it is expected that the stochastic solution might also indicate that sufficient water will be available.

The stochastic problem faces many different situations than the deterministic problem. The two most important of these are the fact that inflows can vary considerably from the deterministic case and that due to the uncertainties involved, the storage of water is a much more significant factor. Recall that in the deterministic case, the reservoirs tended to be held at their minimum levels. For the stochastic problem, this would generally not be the case.

The raw data for this problem was supplied by the Texas Water Development Board (1975) which included the proposed reservoir design, the reservoir capacities, the benefits for meeting the specified demands and the monthly inflow data.

The formulated model is shown in Figure 7-9. Note that this is exactly the same model as each of the 12 periods for the deterministic case.

As in the deterministic case, the total annual demand will be spread evenly over the 12 months of the multiperiod problem. This is reflected in the arc capacities going from the reservoirs or junction nodes to the demand nodes.

Table A-3 in Part II of the Appendix lists the mean inflows, which were used for the deterministic problem, along with their calculated standard deviations. Other data for the Guadalupe River Basin is also shown in Part II of the Appendix. For the stochastic example problems in section 7-1, the inflow data was assumed to be distributed normally. However, a brief survey of the data for this problem reveals that the data is most likely not normal. A Shapiro-Wilk (1965) test for normality was conducted which verified this conclusion. The Shapiro- Wilk W statistic was calculated to be .872. This value would have to be .924 or greater to accept the normal hypothesis at the .01 level of significance.

Many water resources managers are concerned with the

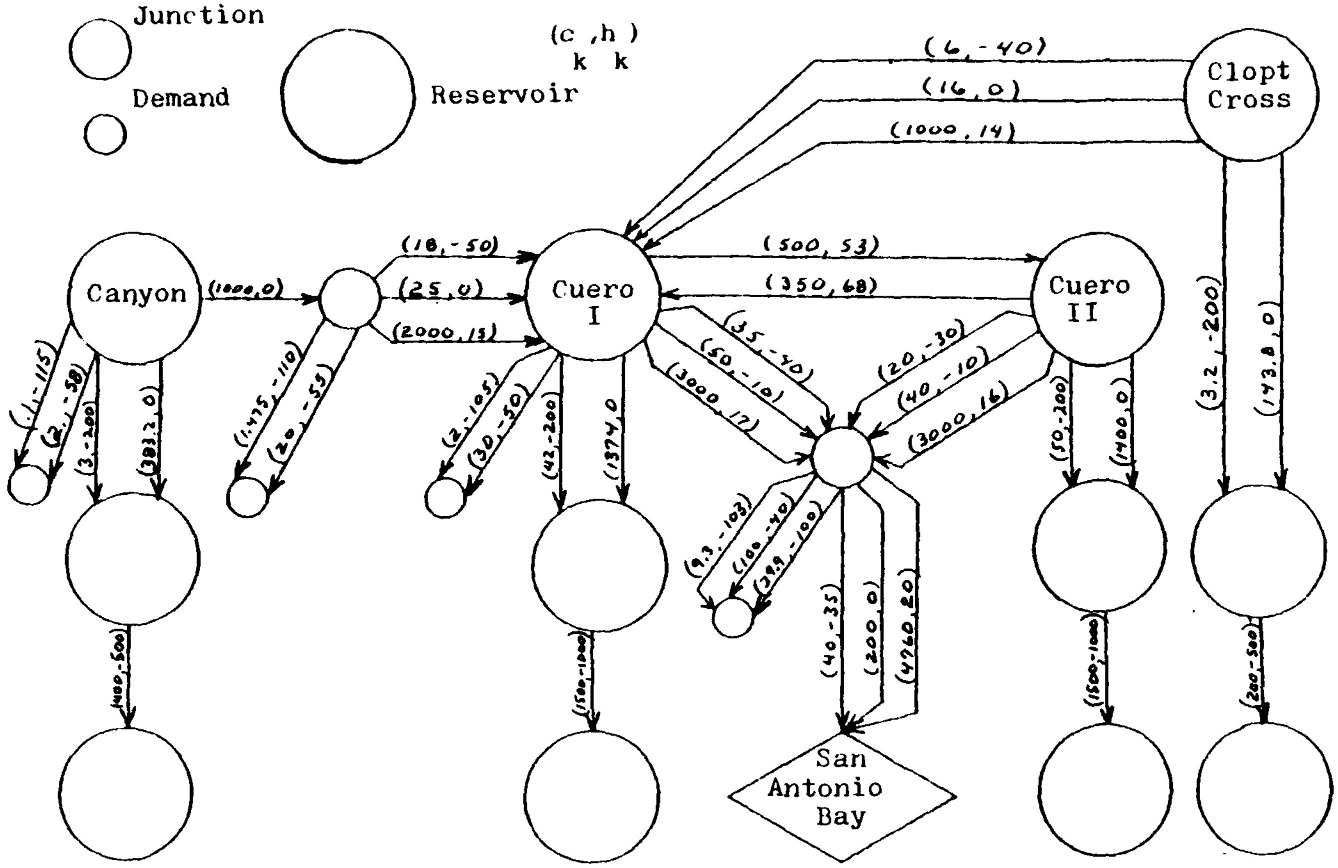


Figure 7-9 Network Model of Guadalupe River Basin

occurances of floods and their analysts typically consider the inflow distribution for floods to be log-normal or in some cases log-normal Pearson type III. Although this problem is concerned with the operation of a system of reservoirs under all inflow conditions, it was decided to test the data to see if it was distributed log-normally. This test had to be altered some since there were several months whose inflows were zero which is not allowed for the log-normal case. Accordingly, the zero inflows were ignored and a test for normality was put to the logarithms of the remaining data. Again the results indicated that this data did not possess the characteristics of a log-normal distributon. The Shapiro-Wilk W statistic was calculated to be .8629.

After several other considerations were ruled out, it was decided to simply use the emperical data as is, and supply the random number sequence to corresponding inflows. Since the data was already arranged for all four reservoirs by month and year, the question of correlating the inflows between reservoirs was taken care of. In the earlier examples the same random number was applied to all reservoirs. Thus, the inflows for all reservoirs in the basin were perfectly correlated. For this problem, if the random number for the month of July turns out to be 38, the inflows for all reservoirs will be determined by selecting the corresponding July inflows in the year 1962 (1925 + 38 - 1). The random number generator will generate a number between 1-46 from a uniform distribution. This number will represent the desired row in the inflow matrix for each of the reservoirs.

One thing to note with regards to this data is the extreme fluctuation of inflows. The most severe fluctuation occurs at the Cuero I reservoir where in the month of July, the range of inflows varied from zero acre-feet in 1963 and 1964 to 648 acre-feet in 1936. Other typical fluctuations range from a low of zero to 200+ acre-feet for some months. This factor alone indicates the severe instability and unreliability of water supplies for this area of Texas.

Based upon the decisions made in Chapter 6, this model was run for 12 periods using 30 draws per level. The discretized range of levels was selected to be (.2,.55,.9). For this four reservoir model there will be 25 level combinations in the experimental design matrix. At 30 draws per level, there will be 750 nonlinear network problems to solve for each period or a total of 9000 for the 12 periods. The Q_T assumed for this problem were as shown in Table 7-9 (col a). The coefficients for the Q_1 benefit function derived using the dynamic programming algorithm are shown in Table 7-9 (col b)

Before discussing the results of this application a few points should be made. First of all, only 46 years of data were available. Ideally, this is not enough data to attempt to characterize the distribution of inflows and hence supports the use of the emperical data. Secondly, the random selection of 30 draws was done with replacement. Finally, observing Table A-4 of the

Table 7-9

Q_T and Q_1 For Guadalupe Basin (All Years)

	a	b
	Q_T	Q_1
B1	-500.	-196.
B2	-500.	-388.
B3	-1000.	-82.
B4	-1000.	-86.
B5	1.0	.16
B6	1.0	1.32
B7	1.0	.011
B8	1.0	.016
B9	0.0	.014
B10	0.0	.019
B11	0.0	.011
B12	0.0	.014
B13	0.0	.007
B14	0.0	.010

Appendix, the projected demands for this problem are extremely low when compared to the quantities of water available. A strict application of this model could be expected to yield sufficient water to supply all demands.

Returning to the results for this problem, note the benefit function for Q_1 is based on the assumption that t = 1 is May. Thus, the last month of the time horizon is April. Since it takes 8-11 months for the effects of the initial conditions to subside, these results are good for only about two months. However, if this model were run for 24 months, the benefit functions for each of the last 12 months would be valid representations of the future value of water.

The results of Table 7-9 indicate that for Canyon and Cloptin Crossing, the initial allocation of water would be to storage since their linear terms are less than any of the demand cost functions. For the two Cuero reservoirs, the top priority is to meet demand. Only Canyon and Cloptin Crossing have significant terms in the quadratic function. As more and more water becomes available these terms will eventually reduce their marginal cost to a level where supplying demand will be profitable. These results are expected since using all inflow data and 30 draws, the expected inflows will approach the mean inflows used in the deterministic case. The design of these reservoirs was such that they would be expected to meet all demands over a 10 year low flow condition given that they were full at the begining of this 10 year period.

Thus, the above results using expected inflows are not surprising at all.

In order to investigate a particularly dry period one may impose upon the system a 10 year low flow profile for inflows. This was done using the years 1947-1956 as the 10 year low flow period. The resulting benefit function, Q_1, for May and again for August are shown in Table 7-10. For the low flow profiles, all of the linear terms of the quadratic are higher than they were when the data for the entire 46 years were used. As before, Canyon and Cloptin Crossing have the lowest linear cost terms with the two Cuero reservoirs marginal costs becoming nearly equal to the demand marginal costs. Also, the terms of the quadratic, while slightly more pronounced, are similar to the all years results. Again, for both cases, the results appear to indicate that the design is sufficient to satisfy the 10 year low flow profile. This is not surprising since the design was selected to meet this criteria.

The main conclusion to be drawn from this application is that the reservoirs seem to be adequately designed (if not overly designed) to satisfy their intended purpose of safegarding against shortages in the year 2020. The derived benefit functions could be used to make operational decisions if this system were built.

7.3 Number and Duration of Time Periods

The selection of the number and length of time periods is highly dependent upon the specific problem being considered.

Table 7-10

Q_1 Results for May and August (Guadalupe Basin)

	Q_1 May	Q_1 August
B1	-340.	-353.
B2	-536.	-550.
B3	-106.	-115.
B4	-98.	-103.
B5	.332	.341
B6	1.70	1.74
B7	.017	.019
B8	.017	.018
B9	.012	.013
B10	.028	.030
B11	.012	.013
B12	,010	.014
B13	.005	.006
B14	.012	.013

Consequently, this section is intended to point out some of the possibilities and concerns that the user should be aware of. The sample problems in this chapter dealt with some of the specifics alluded to in this section.

It first must be determined what types of decisions the user needs to make. Is he concerned with long term or short term information? In the long term case he may want to let a time period be 3 months or as long as a year and consider a time horizon of 20 years. This might be the case if he were concerned with using the results of the model for the evaluation of the design of a proposed new system.

On the other hand, if he is more concerned with the current operation of a system, he may want to specify his time period as a month or perhaps a week and run his model for from 1 to 5 years. In either case, he may also be limited by the available data. Specifically, he will require access to historical runnoff data which may not be available for a given choice of time duration or horizon.

These two options (and there are several others) require different sets of data and assumptions. For instance, in the case of current operations, he would be much more concerned with the accuracy of the model parameters such as the benefits for supplying demand and would require better knowledge of the inflow statistics.

Another concern might depend upon the nature of droughts in the basin of concern. Some parts of the country are more sensitive to short term droughts due to their immediate effect on agriculture or livestock whereas others may experience droughts which may last for several years.

In any case, he would like to select a time horizon that is far enough into the future such that the initial estimates of the quadratic benefit function for period T has little effect upon the functions derived for the early periods. For most of the trial data, and for the sample problems a duration of one month was considered and the time horizon was restricted to one year.

CHAPTER VIII

8. Summary

A review of the literature indicates that although much work has been done in determining optimal water resources management policies, only in the last 5-10 years have authors begun to address larger systems to include stochastic inflows and interactive benefits realized for a multireservoir system. This is important for many reasons. First of all of primary concern in the development of a realistic water resources model is the inclusion of the future uncertainties of water supplies. Because of this uncertainty, current decisions regarding the storage or release of water have a direct effect on future operations. Provisions must be allowed for continued operations in the event of future shortages or oversupply.

To evaluate the system given future uncertainties, it is felt that through the optimal operation of a multireservoir system, greater benefits can be realized than through the individual optimization of each reservoir. This joint benefit stems from the ability to transfer water between some reservoirs in the system and creates the need for a means of representing and evaluating these interactive effects. This is handled by generating a nonlinear nonseparable quadratic representation for the future value of water.

One of the principle contributions of this research has been the formulation and computer implementation of an optimization algorithm for solving the convex nonlinear, nonseparable minimization problem for the generalized network. It is believed that this work represents the first application of nonlinear generalized network codes modified to handle such problems, without reverting to a piecewise linear approximation. This modification builds upon the highly efficient linear codes of Jensen and Barnes (1980) and extends the practical application of network codes to a new set of problems. The only restriction for this problem is that the overall objective function which is a combination of several linear terms and some quadratic terms be a convex cost function.

Another major contribution of this research is the formulation and computer implementation of a stochastic dynamic programming algorithm for solving multiperiod decision problems with uncertainty. This algorithm combines the techniques of dynamic programming, network flow programming and regression analysis in a unique way. A principle feature is the representation of the value of the recursive function as a nonlinear function. This functional representation greatly relieves the dimensionality problems usually associated with large dynamic programming problems. This benefit function is derived by using the network programs to optimize the model. These optimal results are then used in a least squares regression model which fits a full guadratic to the data. This benefit function is then translated into network parameters and these become known values for the solution of the network in the next dynamic programming period. By following the typical dynamic programming recursion methods, the benefit functions for each period are ultimately generated and can then be used to evaluate decisions in an operational context.

Using the above mentioned algorithm, these procedures are applied to a water distribution problem. Several example applications using hypothetical systems are evaluated to verify and demonstrate the approach. An application is then made to the Guadalupe River Basin in Texas. This application involves a proposed four reservoir system for this basin designed to satisfy the projected demands of the year 2020. The results showed the Texas Water Development Board four reservoir design to be more than adequate to meet the projected demands.

One of the primary advantages of this model formulation for the water resources application is the ease of providing the required data. Since the multiperiod model is simply multiple copies of the single period model, only data relating to changes need be provided for each new period. The program automatically adjusts the network parameters to account for the newly derived quadratic form. Thus, the only new data that needs to be supplied is the changing inflow parameters. These are simply read in for each new period as the model progresses through time. Great flexibility is allowed here since the user can model any inflow profile he chooses. Periods of long drought are often of concern to water resources managers and this can easily be modeled. Inflow data can be provided to represent the 10 year low flow condition for the given basin. Alternative basin designs can be evaluated under similar rainfall conditions. Changing demands due to seasonal requirements or due to economic pressures can also be modeled.

Another very important "flexibile" capability allows the user to stop the process at any time and to restart from that point at a later time. This is due to the functional expression for the benefit function which is stored in quadratic form and can very easily become the starting point for a later trial.

The flexibility of this model is not limited to the items mentioned above. There are a myriad of factors that can easily be changed to allow the user to model almost any situation desired.

The entire process of using a generalized network model adapted to solve quadratic forms, regression analysis to derive a functional expression for the future value of water and dynamic programming to model the multiperiod decision process represents a unique combination of these techniques as applied to the multiperiod multireservoir water resources stochastic problem. Using the algorithms and methodologies derived in this research it is felt that this model can have several practical applications in areas other than water resources.

APPENDIX

This appendix is divided into three parts as indicated below:

Part I Data sets for the example problems

Part II Guadalupe River Basin

Part III Flow charts

Part I

Data Sets for the Hypothetical Problems

Tables A-1 and A-2 show the actual inflows used for the hypothetical problems. Table A-1 shows the inflow data for the three reservoir problem for inflow profiles a, b and c of Figure 7-4 respectively. Table A-2 shows similar data for the four reservoir problem.

The network parameters for these examples were shown in Chapter 7.

Table A-1

3 Reservoir Inflow Data

	Profile a		Profile b		Profile c	
Month	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev
	5.0	5.3	1.8	1.5	3.0	1.5
12	3.3	3.0	1.6	1.2	2.5	1.8
	2.2	1.8	1.0	.9	3.2	1.0
	7.2	6.0	2.3	2.0	4.0	2.0
11	6.0	5.0	1.8	1.6	4.2	2.1
	5.0	4.0	1.4	1.1	4.4	2.2
	7.5	6.5	3.0	2.5	5.0	2.5
10	6.0	5.0	2.8	2.2	4.9	2.5
	4.0	2.7	1.5	1.3	5.2	2.0
	6.8	6.0	4.0	3.2	6.0	2.8
9	5.2	4.5	3.2	2.6	6.3	2.7
	3.0	2.5	1.6	1.0	6.7	3.0
	6.0	5.5	4.1	3.2	7.5	3.2
8	4.0	3.0	3.3	3.0	8.0	3.9
	3.5	2.6	3.0	2.6	7.0	3.4
	5.2	4.8	4.2	4.0	6.0	3.0
7	4.2	3.4	3.5	3.3	5.8	2.9
	3.0	2.5	3.2	2.9	6.0	3.0
	4.2	4.0	5.2	4.8	5.0	2.5
6	3.5	3.3	4.2	3.4	4.2	2.1
	3.2	2.9	3.0	2.5	5.2	2.6
	4.1	3.2	6.0	5.5	4.0	2.0
5	3.3	3.0	4.0	3.0	3.4	1.7
	3.0	2.6	3.5	2.6	4.2	2.1
	4.0	3.2	6.8	6.0	3.0	1.5
4	3.2	2.6	5.2	4.5	2.5	1.4
	1.6	1.0	3.0	2.5	3.2	1.6
	3.0	2.5	7.5	6.5	8.0	4.0
3	2.8	2.2	6.0	5.0	9.4	4.7
	1.5	1.3	4.0	2.7	7.8	3.9
	2.3	2.0	7.2	6.0	12.5	6.3
2	1.8	1.6	6.0	5.0	11.9	5.9
	1.4	1.0	5.0	4.0	13.2	6.6
	1.8	1.5	5.0	5.3	15.0	7.5
1	1.6	1.2	3.3	3.0	12.8	6.4
	1.0	.9	2.2	1.8	14.0	7.0

Table A-2

4 Reservoir Inflow Data

	Profile a		Profile b		Profile c	
Month	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev
12	15.0	7.5	4.0	2.0	4.0	2.0
	18.0	9.0	2.0	1.0	2.0	1.0
	12.0	6.0	3.0	1.5	3.0	1.5
	24.0	12.0	2.5	1.25	2.5	1.25
11	13.5	7.0	6.0	3.0	6.0	3.0
	14.5	7.3	3.0	1.5	3.0	1.5
	11.0	5.5	4.2	2.1	4.0	2.0
	23.0	11.5	3.6	1.8	4.0	1.9
10	12.4	6.4	8.0	4.0	8.0	4.0
	13.8	6.9	4.0	2.0	4.0	2.0
	10.0	5.0	5.4	2.7	6.0	3.0
	21.0	10.5	4.7	2.3	5.0	2.5
9	11.3	5.4	10.0	5.0	10.0	5.0
	12.6	6.3	4.6	2.3	5.0	2.5
	9.0	4.5	6.6	3.3	7.5	3.75
	20.0	10.0	5.8	2.9	6.2	3.1
8	10.2	5.1	12.0	6.0	12.0	6.0
	11.4	5.7	5.0	2.5	6.0	3.0
	8.0	4.0	7.8	3.9	9.0	4.5
	18.0	9.0	6.9	3.5	7.5	3.75
7	9.1	4.6	14.0	7.0	10.0	5.0
	10.2	5.1	6.0	3.0	5.0	2.5
	7.0	3.5	9.0	4.5	7.5	3.75
	16.0	8.0	8.0	4.0	6.2	3.1
6	8.0	4.0	16.0	8.0	8.0	4.0
	9.0	4.5	7.0	3.5	4.0	2.0
	6.0	3.0	10.2	5.1	6.0	3.0
	14.0	7.0	9.1	4.6	5.0	2.5
5	6.9	3.5	18.0	9.0	6.0	3.0
	7.8	3.9	8.0	4.0	3.0	1.5
	5.0	2.5	11.4	5.7	4.0	2.0
	12.0	6.0	10.2	5.1	4.0	2.0
4	5.8	2.9	20.0	10.0	4.0	2.0
	6.6	3.3	9.0	4.5	2.0	1.0
	4.6	2.3	12.6	6.3	3.0	1.5
	10.0	5.0	11.3	5.4	2.5	1.25

Table A-2 (continued)

3	4.7	2.3	21.0	10.5	10.0	5.0
	5.4	2.7	10.0	5.0	6.0	3.0
	4.0	2.0	13.8	6.9	8.0	4.0
	8.0	4.0	12.4	6.2	7.0	3.5
2	3.6	1.8	23.0	11.5	16.0	8.0
	4.2	2.1	11.0	5.5	9.0	4.5
	3.0	1.5	14.5	7.3	13.0	6.5
	6.0	3.0	13.5	7.0	11.0	5.5
1	2.5	1.25	24.0	12.0	24.0	12.0
	3.0	1.5	12.0	6.0	12.0	6.0
	2.0	1.0	18.0	9.0	18.0	9.0
	4.0	2.0	15.0	7.5	15.0	7.5

Part II

Guadalupe River Basin

In Chapter 3 the problem of water distribution for the Guadalupe River Basin, deterministic case, was discussed. The basic determiistic network for each period is shown in Figure A-1. The inflow data and results are shown here.

The single period model of Figure A-1 for this problem remains the same for all 12 periods. For this particular example, all arc parameters also remaine the same for all periods. This means, for instance, that the benefit for demand and the amounts demanded are equal in all periods. This will most likely not be the case in a realistic situation and can easily be changed.

Table A-3 lists the deterministic inflows (the means) along with their calculated standard deviation (which are not used in the deterministic case) for each of the four reservoirs. These deterministic inflows are the fixed external flows for the reservoirs in the corresponding period. For the 12 period model, the inflow data from 1925-1970 was averaged by month to yield these results.

The raw data for this problem was provided by the Texas Water Development Board. They previously adjusted this data to reflect the actual expected inflows for these reservoirs over this time horizon given that they had been in existence. This same raw

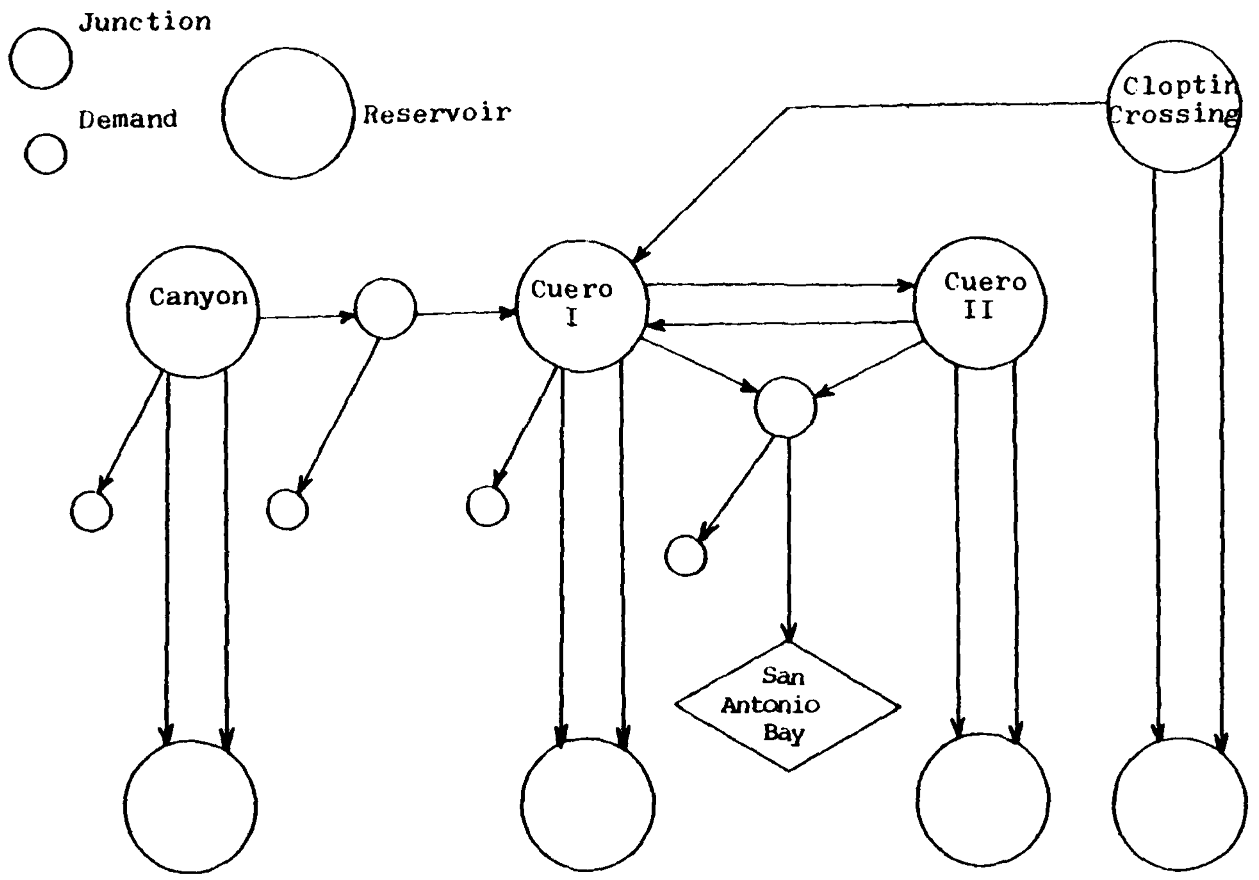


Figure A-1 Guadalupe River Basin - Deterministic Case

Table A-3

Guadalupe Basin Inflows By Reservoir

	Canyon		Cloptin Crossing	
Month	Mean	Std.Dev	Mean	Std.Dev
1	18.04	23.17	5.87	9.85
2	18.87	20.45	6.67	8.86
3	20.48	19.73	7.00	7.67
4	23.02	23.07	8.65	10.01
5	32.09	33.00	11.33	15.53
6	24.67	37.80	7.22	8.64
7	17.26	34.84	3.98	5.09
8	7.98	9.61	2.07	1.57
9	20.28	40.62	4.93	11.23
10	20.48	26.68	4.43	7.29
11	12.74	13.93	3.70	5.30
12	14.61	14.21	4.50	5.87

	Cuero I		Cuero II	
Month	Mean	Std.Dev	Mean	Std.Dev
1	30.48	42.86	5.07	10.06
2	33.15	47.70	5.91	9.57
3	30.74	26.98	4.41	6.91
4	50.39	63.86	6.43	12.40
5	64.02	87.43	12.33	20.15
6	42.67	49.90	12.87	20.39
7	37.93	101.97	3.11	4.28
8	13.76	35.50	2.00	3.66
9	24.80	47.49	13.04	53.24
10	27.22	48.37	13.26	45.09
11	28.33	47.52	3.59	7.60
12	23.98	31.77	2.04	3.10

data will be used later for the stochastic case.

The objective for this problem is twofold. The first is to determine if enough water will be available on the average to meet the projected demands for the year 2020. Secondly, what will be the distribution of water for each of these periods, ie., what is the decision set?

Accordingly, the capacity of the demand arcs are set to 1/12 of the total annual demand for the year 2020 for each demand location.

Figures A-2, (a,b,c,d) represent the 12 copies of this single period model. These 12 periods are linked together and the last period is linked back to the first. To run this problem it was necessary to give each of the reservoirs an initial level of water. To approach this problem from a worst case position, each of the reservoirs was given an initial level of just 5 units of water.

Table A-4 shows the capacity of the proposed reservoirs, their initial conditons, their minimum requirements and the annual demands placed upon each demand point. The auxillary demands listed are intended to account for the possibility of reducing the reliance on te Edwards Aquifer by supplying more water from the reservoir system. For this problem, the period 1 data represents January and period 12 represents December of the year 2020. Note that since this is a linked problem any month could be used as the starting point and run for 12 periods. The results would have been

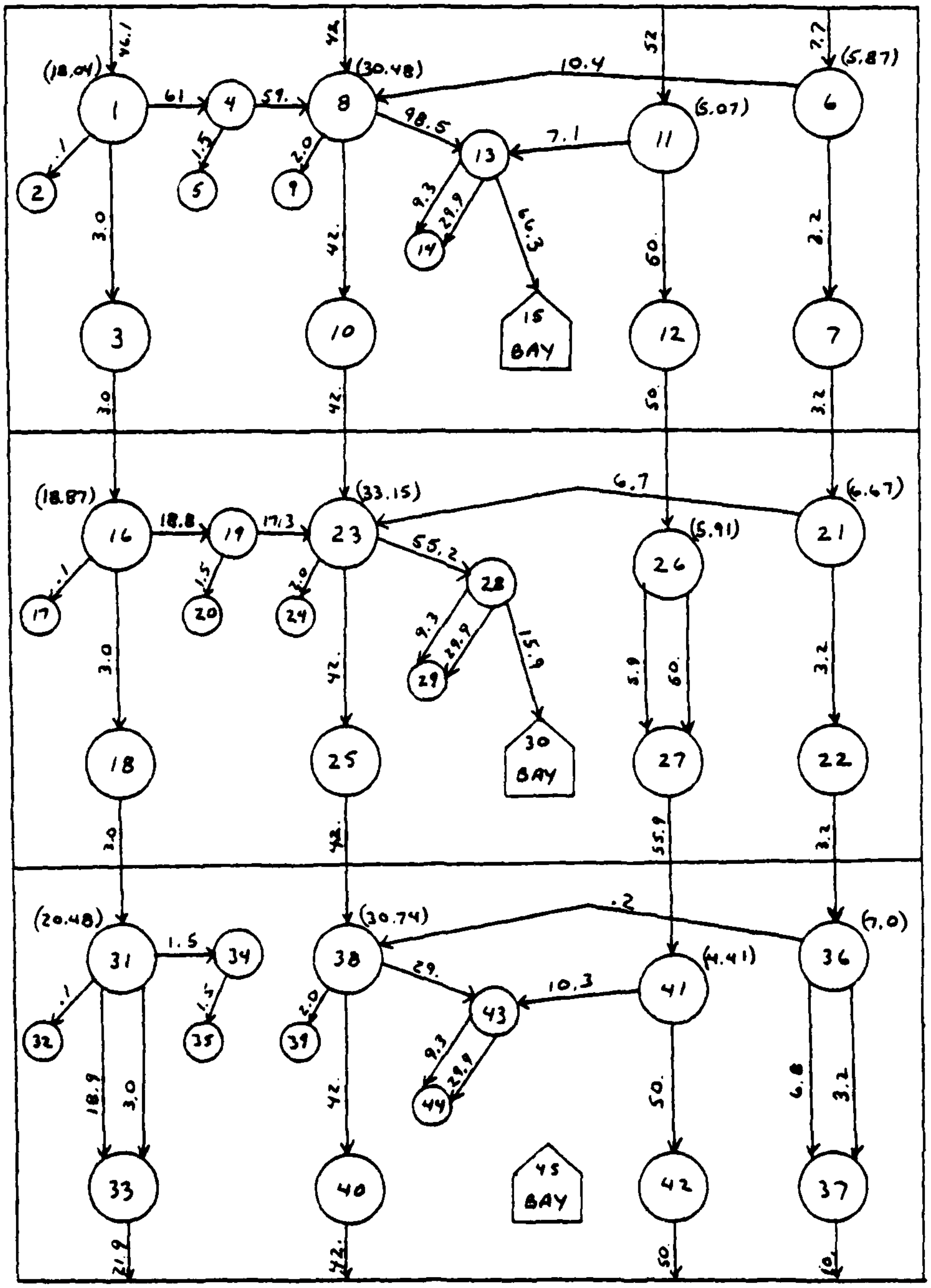


Figure A-2 12 Period Deterministic Solution With Flows

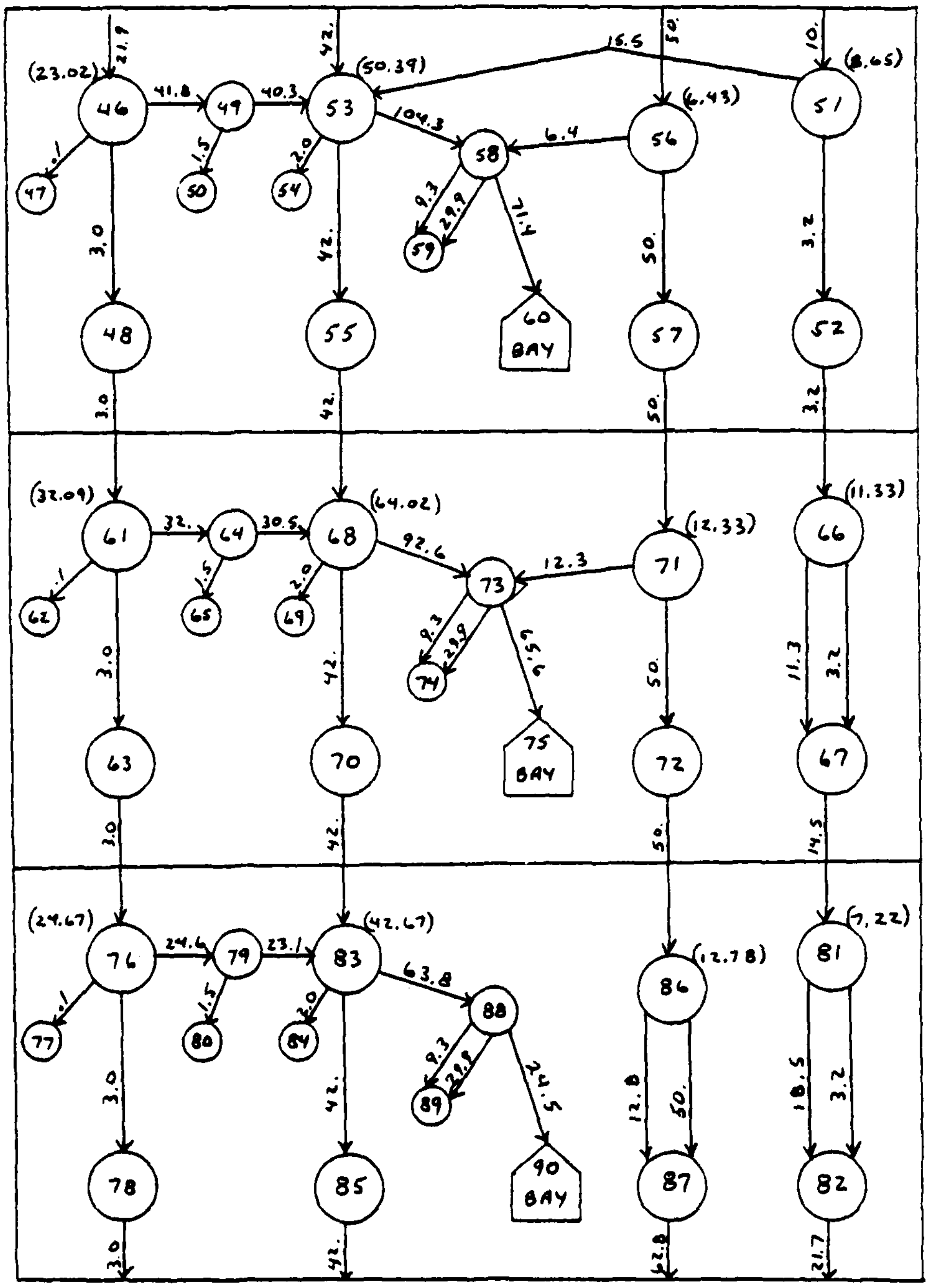


Figure A-2 Continued

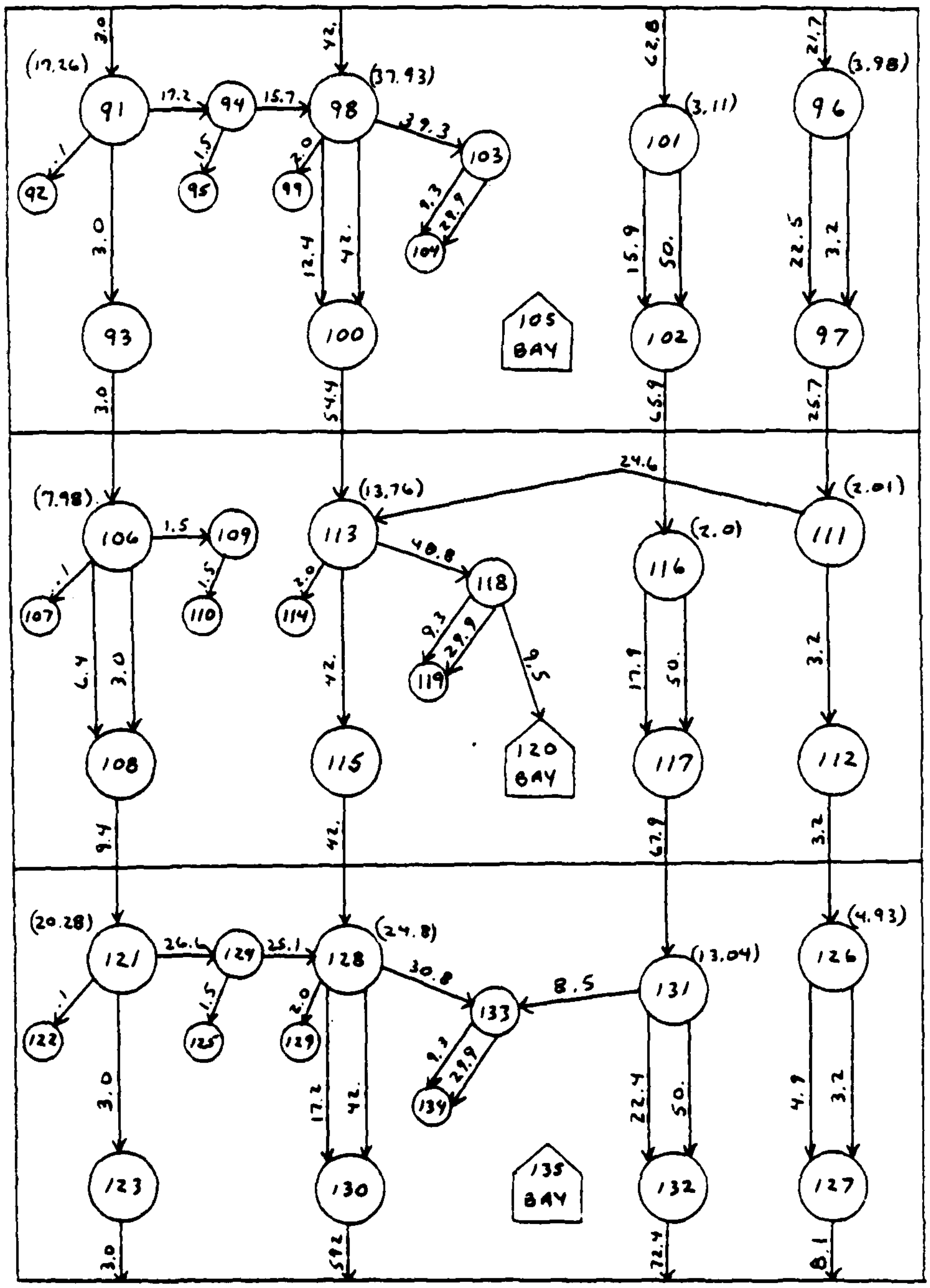


Figure A-2 Continued

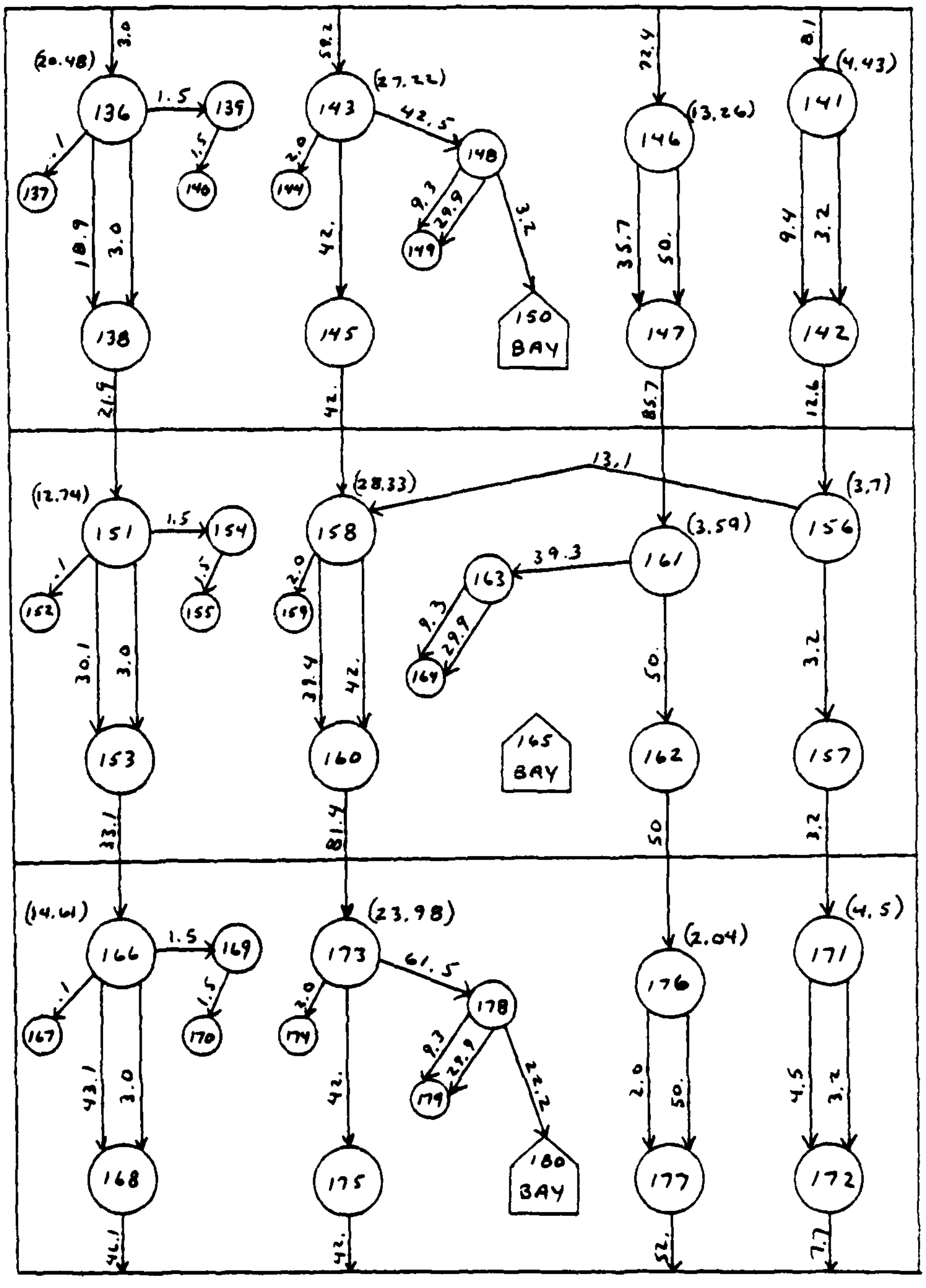


Figure A-2 Cortinued

Table A-4

Guadalupe River Basin Capacities
(1000 acre feet)

Basin or Junction	Capacity	Initial Conditions	Minimum Req'mts	Annual Demands	Auxillary Demands
Canyon	386.2	5.0	3.0	2.0	24.0
Cloptin Crossing	147.0	5.0	3.2	---	---
Cuero I	1416.	5.0	42.0	23.5	360.
Cuero II	1450.	5.0	50.	---	---
Seguin	---	---	---	17.7	240.
Victoria	---	---	---	471.3	1200.
Totals	3399.2		98.2	513.5	1824

the same.

The results indicate that enough water will be available on the average to meet all demands for all periods. Since there was no penalty assessed for releasing water into San Antonio Bay, and no reward for building up levels, the reservoirs tended to be held at their minimum levels. Given no penalty or reward for doing otherwise, this is what one would expect with known fixed inflows and demands.

The same input data was used for the stochastic application to the Guadalupe River Basin. In Figure 7-9 the stochastic (nonlinear) single period network model was shown with all arc parameters specified. For this problem, use of the mean and standard deviation would imply that the raw inflow data could be characterized by some probability distribution. Since this was determined to be infeasible, the emperical data was used. The emperical data was provided by the Texas Water Development Board.

For the month in question, a random number was drawn which corresponds to the inflows for a given year. The inflows for the other three reservoirs were then chosen to be from that same year, thus correlating all reservoirs for this basin.

Part III

Flow Charts

This part of the appendix includes the following:

A. A schematic of the 13 special or new programs which are required for this problem showing the general relationships between them. (Figure A-3)

B. A brief description of the 13 programs of the schematic.

C. Flow charts for most of the programs shown in the schematic. Only the logic required for determining the benefit functions using the ordinary least squares solution methodology is flow charted. The logic required to calculate weighted least squares coefficients and many of the other statistics that were determined in this report is not shown on the flow charts. This logic is however, still in the computer program for future access.

Part A

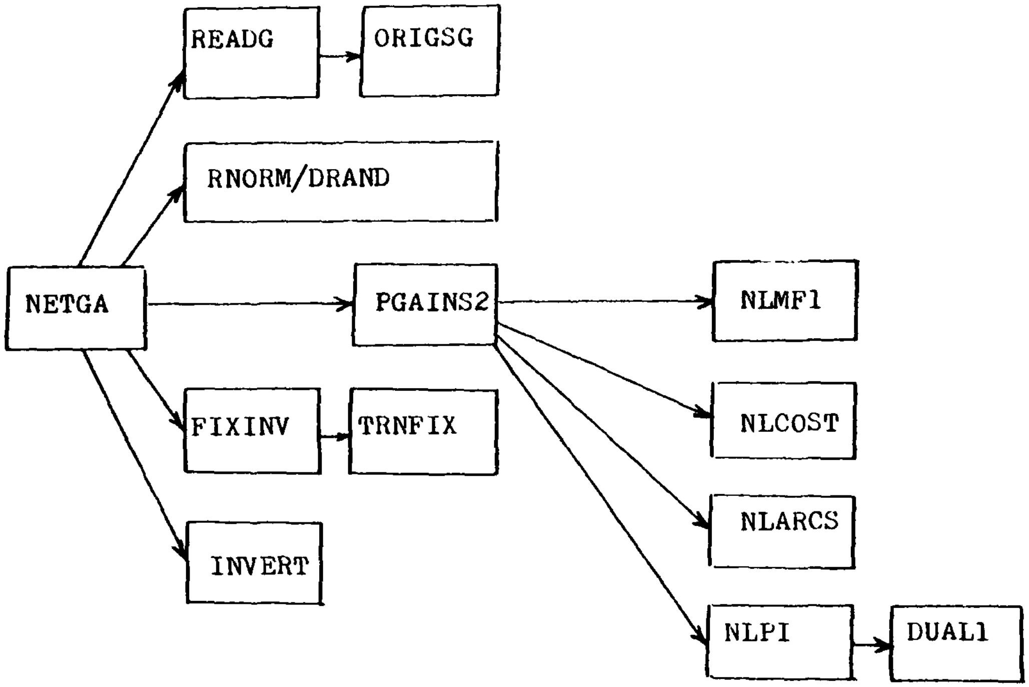


Figure A-3

Schematic of Special Programs

Part C

Brief Description of Flow Charts

NETGA This is the main program for this problem. This program calls the others as shown in schematic B. It also reads some of the reservoir and runoff data, sets up the design matrix of reservoir levels and makes the adjustments for the linear arcs and Q matrix. Also, all calculations for the benefit function coefficients for both ordinary and weighted least squares are done here. The statistical data is mostly computed in this main program.

READG This subroutine reads the network data. It has been modified to read the reservoir data as a part of the node input. The nonlinear arc information is read here as part of the arc data. This includes a linear cost term for the full quadratic. The all artifical arc basis is set up here, and the Q matrix is read.

ORIGSG Revised only to account for the nonlinear arcs.

RNORM RNORM and DRAND are functions used to derive the random numbers. RNORM is used to return a random number from a normal distribution with zero mean and a standard deviation of 1. DRAND is used to return a random number between zero and one from a uniform distribution. These functions are not flow charted in Part C.

FIXINV This subroutine performs all the transformations on the input data to define the necessary terms for the full quadratic design matrix.

TRNFIX This subroutine transforms the three input variables into the full 9 variable quadratic: (the 10th term is the constant which is added to this list in FIXINV). The logic herein is general and will perform the required transformations for a full quadratic given any number of variables.

INVERT This subroutine performs all required matrix inversions. For the selected methodology it is only called from NETGA. However, for calculating the WLS data, it is also called from FIXINV. This program is not flow charted in Part C.

PGAINS2 This is the routine that masterminds the solution process of the network. It is a modification of the subroutine PGAINS which solves the network with gains problem using the primal approach.

NLMF1 This subroutine is called by PGAINS2 to determine the maximum flow change allowed in the nonlinear arcs which causes its marginal cost to be zero. This amount is returned to PGAINS2 and compared with MF which is the maximum flow change allowed by the linear arcs. The appropriate arc is deleted from the basis through the use of the usual network subroutines and an entering arc is selected.

NLCOST This subroutine assists in the above process by determining the cost to be associated with the nonlinear arcs. By knowing the costs that are attributed to all arcs, the entering arc can be selected. These costs are also needed for the PI update.

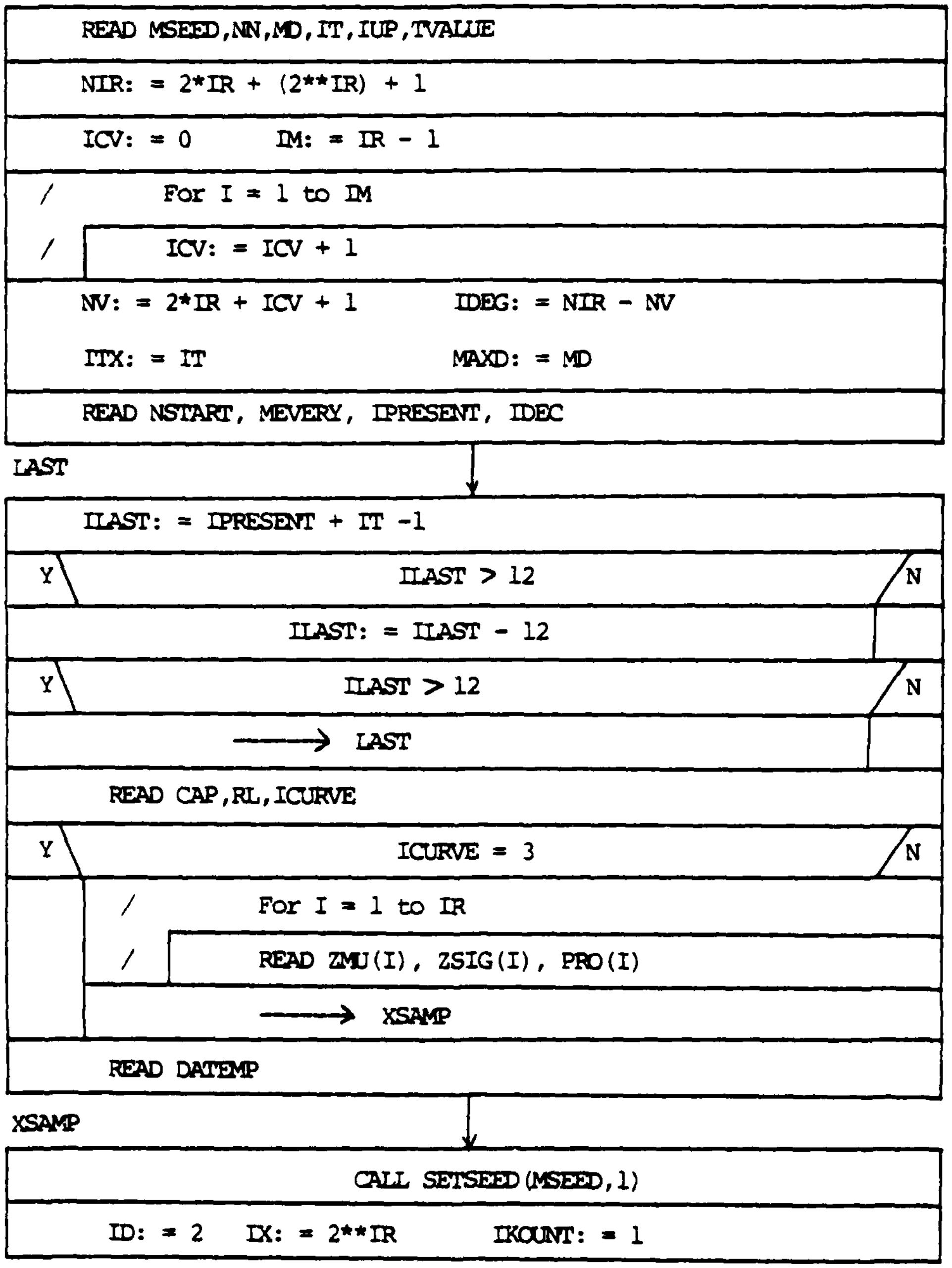
NLARCS This subroutine examines the basis after the tree has been updated to determine if and how many nonlinear arcs are in the basis. If there are no nonlinear arcs in the basis, PGAINS2 calls DUAL (an existing subroutine) to update the PI values in that part of the tree rooted at the terminal node of the entering arc. However, if there are any nonlinear arcs in the basis, then all PI values rooted at the origin of the nonlinear arcs, as well as those beyond the entering arc must be updated. In this case, PGAINS2 calls NLPI which in turn calls DUAL1.

NLPI This subroutine is used to specifically find the nonlinear arcs that are in the basis and to appropriately flag them for use in the PI update.

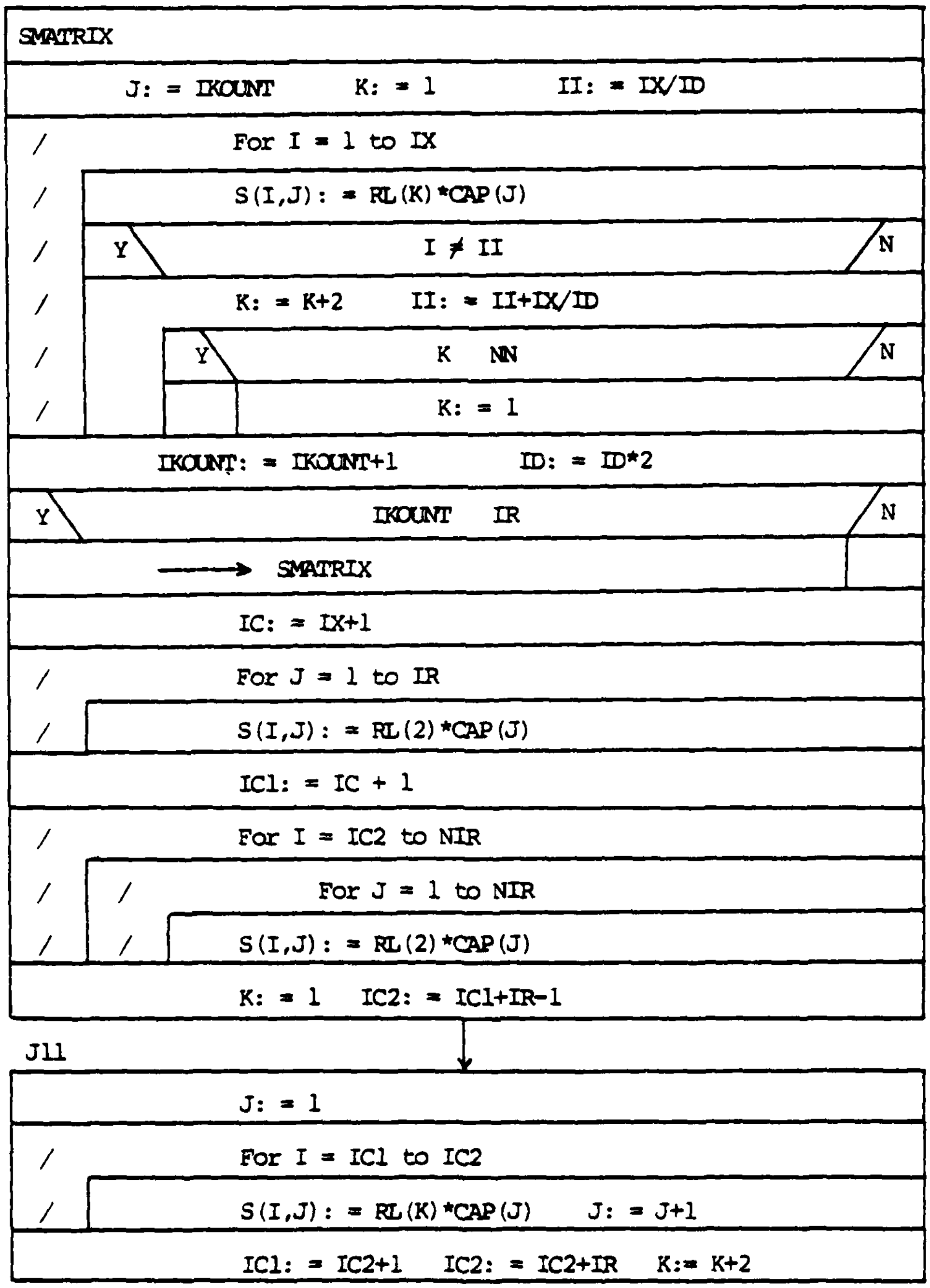
DUAL1 This subroutine is used in lieu of DUAL for the PI update in the presence of nonlinear arcs.

All of the above programs with the exception of RNORM/DRAND and INVERT are flow charted on the pages that follow.

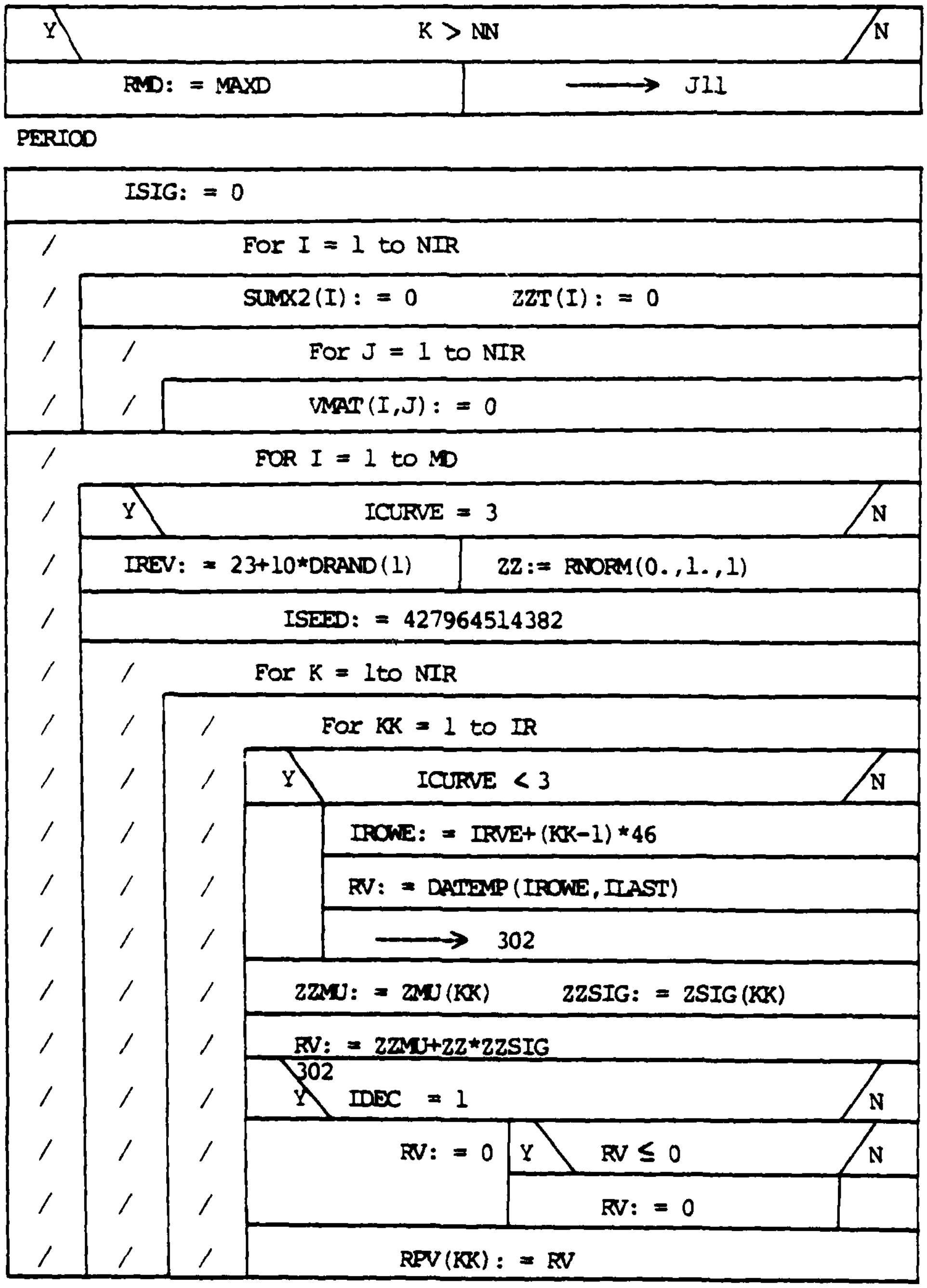
```
NETGA
EPNEG: = -.0001        EPPOS: = .0001
CALL READG
PB(N): = 0    PI(N): = 0    IMAT: = 0    IUPDATE: = 0
CALL ORIG(N,LISA,LISN,L)
Y  L = 0  N
   Y:
   N:
      For KK = 1 to L
         K: = LISA(KK)        J: = T(K)
         For II = 1 to IR
            Y  J ≠ IRESB(II)  N
               Y:
               N: ISUBK(II): = K
         Y  H(K) < 9998.  N
            Y:
            N: PB(J): = K        PI(J): = 9999.
CALL TERM
Y  L = 0  N
   Y:
   N:
      For KK = 1 to L
         K: = LISA(KK)        J: = O(K)
         For II = 1 to IR
            Y  J ≠ IRESB(II)  N
               Y:
               N: IADDK(II): = K
         Y  H(K) < 9998.  N
            Y:
            N: PB(J): = -K    PI(J): = -9999.
CALL TREINT(N)
IT: = 6
CALL PGAINS2(ITER,IT)
```



```
READ MSEED,NN,MD,IT,IUP,TVALUE
NIR: = 2*IR + (2**IR) + 1
ICV: = 0        IM: = IR - 1
/   For I = 1 to IM
/       ICV: = ICV + 1
NV: = 2*IR + ICV + 1        IDEG: = NIR - NV
ITX: = IT                   MAXD: = MD
READ NSTART, MEVERY, IPRESENT, IDEC
LAST
ILAST: = IPRESENT + IT -1
Y   ILAST > 12   N
    ILAST: = ILAST - 12
Y   ILAST > 12   N
    ——> LAST
READ CAP,RL,ICURVE
Y   ICURVE = 3   N
    /   For I = 1 to IR
    /       READ ZMU(I), ZSIG(I), PRO(I)
    ——> XSAMP
READ DATEMP
XSAMP
CALL SETSEED(MSEED,1)
ID: = 2    IX: = 2**IR        IKOUNT: = 1
```



```
SMATRIX
J: = IKOUNT      K: = 1          II: = IX/ID
For I = 1 to IX
    S(I,J): = RL(K)*CAP(J)
    Y    I ≠ II    N
        K: = K+2      II: = II+IX/ID
        Y    K    NN    N
            K: = 1
IKOUNT: = IKOUNT+1          ID: = ID*2
Y    IKOUNT    IR    N
    ——→ SMATRIX
IC: = IX+1
For J = 1 to IR
    S(I,J): = RL(2)*CAP(J)
IC1: = IC + 1
For I = IC2 to NIR
    For J = 1 to NIR
        S(I,J): = RL(2)*CAP(J)
K: = 1    IC2: = IC1+IR-1
J11
J: = 1
For I = IC1 to IC2
    S(I,J): = RL(K)*CAP(J)     J: = J+1
IC1: = IC2+1    IC2: = IC2+IR    K:= K+2
```

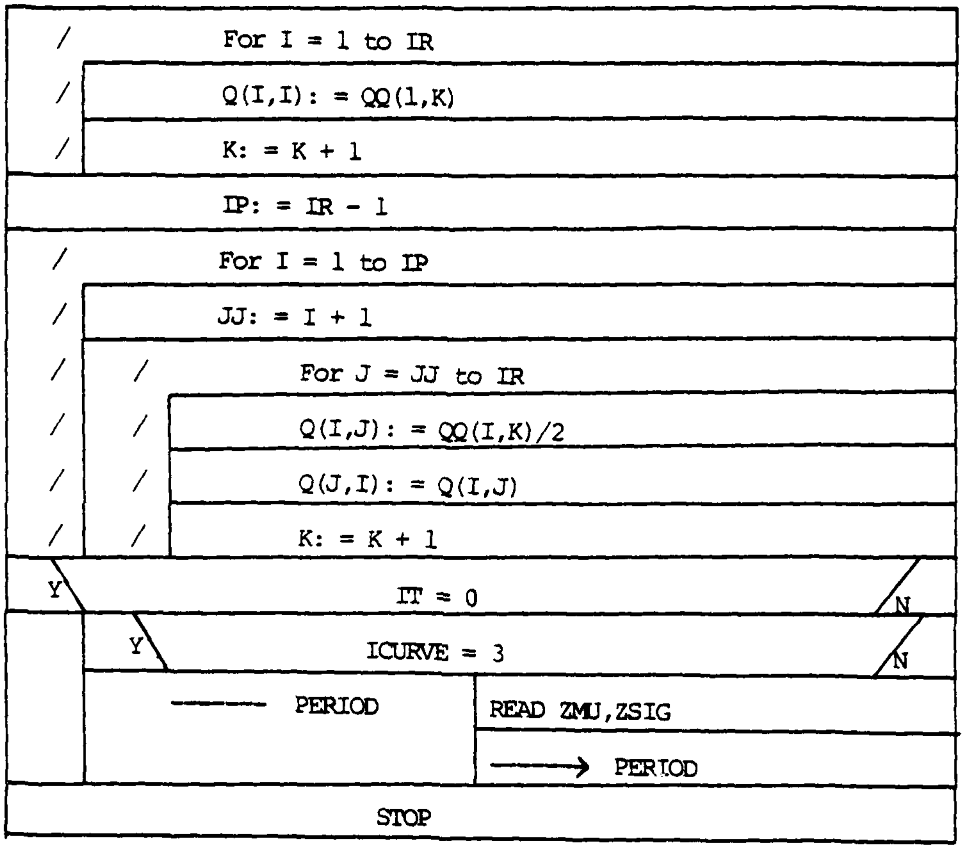


```
IF K > NN
  Y: RMD: = MAXD
  N: ---> J11

PERIOD

ISIG: = 0
For I = 1 to NIR
    SUMX2(I): = 0        ZZT(I): = 0
    For J = 1 to NIR
        VMAT(I,J): = 0
FOR I = 1 to MD
    IF ICURVE = 3
      Y: IREV: = 23+10*DRAND(1)
      N: ZZ:= RNORM(0.,1.,1)
    ISEED: = 427964514382
    For K = 1to NIR
        For KK = 1 to IR
            IF ICURVE < 3
              Y:
              N: IROWE: = IRVE+(KK-1)*46
                 RV: = DATEMP(IROWE,ILAST)
                 ---> 302
            ZZMU: = ZMU(KK)       ZZSIG: = ZSIG(KK)
            RV: = ZZMU+ZZ*ZZSIG
      302   IF IDEC = 1
              Y: RV: = 0
              N: IF RV ≤ 0
                   Y: RV: = 0
                   N:
            RPV(KK): = RV
        ISIG: = ISIG+1
        For J = 1 to IR
            AMS(J): = RPV(J) + S(K,J)
            DIFF: = AMT(J) - AMS(J)
            DIFF = 0
            Y:
            N:
                DIFF > 0
                Y:
                    F(ISUBK(J)): = DIFF
                    C(ISUBK(J)):=DIFF
                N:
                    F(IADDK(J)):=-DIFF
                    C(IADDK(J)):=-DIFF
                AMT(J): = AMS(J)
        CALL PGAINS2(ITER,IT)
        IDEC ≠ 1
        Y:
        N:
            STOP
        ZT(K):=ICOST    SUMX2(K):=SUMX2(K)+ZT(K)**2
        ZZT(K):=ZZT(K)+ZT(K)    ZZZT(I,K): = ZT(K)
    For IV = 1 to NIR
        For JV = 1 to NIR
            VMAT(IV,JV):=ZT(IV)*ZT(JV)+VMAT(IV,JV)
PMD: = FLOAT(MD)
For I = 1 to NIR
    ZZT(I): = ZZT(I)/PMD
For I = 1 to NIR and J = 1 to NIR
    VMAT(I,J):=(VMAT(I,J)-PMD*ZZT(I)+ZZT(J))/((PMD-1)*PMD)
```

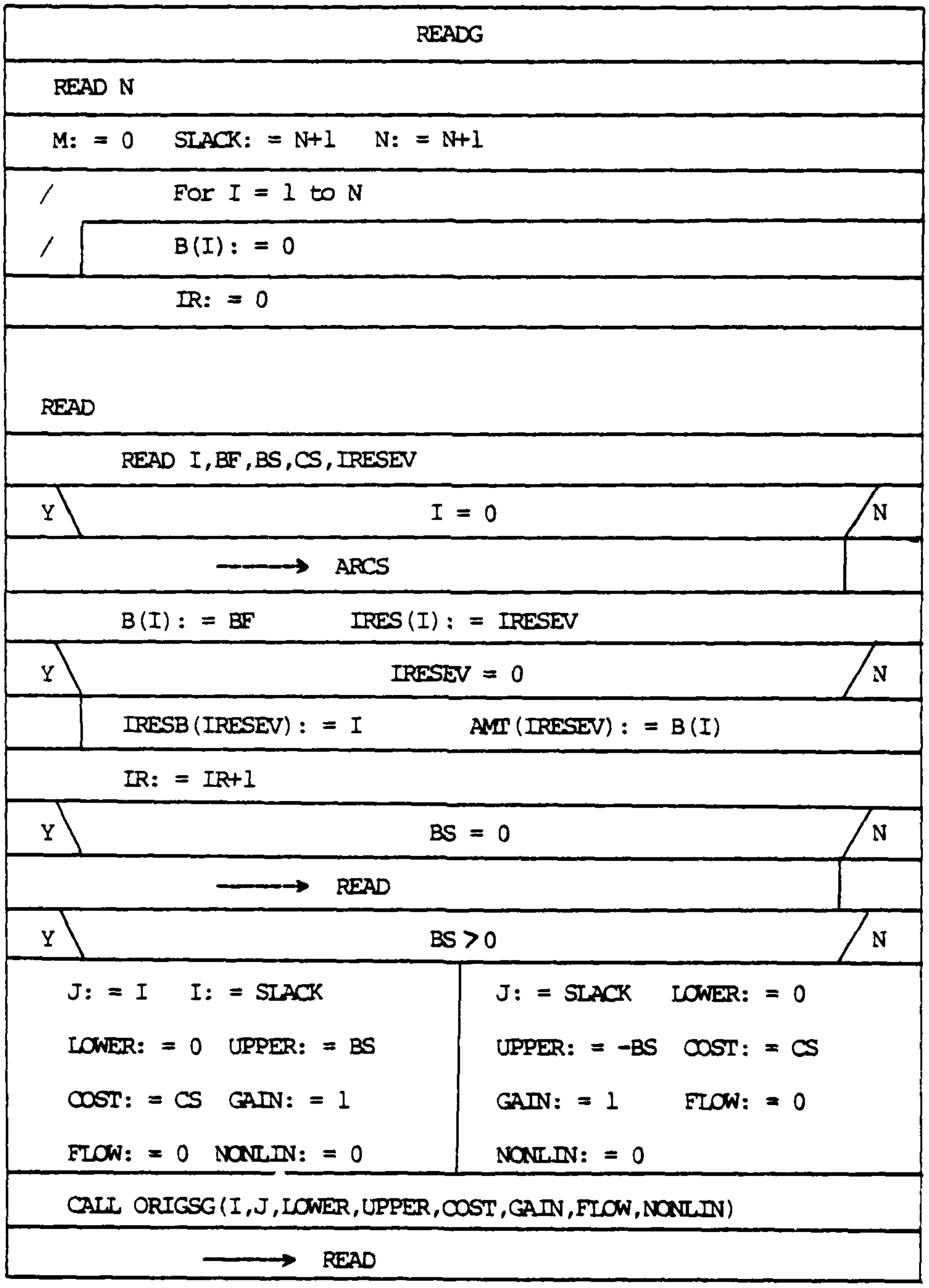
```
For I = 1 to NIR
    For J = 1 to NIR
        VMATOR(I,J): = VMAT(I,J)
CALL FIXINV(IR)
IUPDATE: = IUPDATE+1
For K = 1 to NV
    QQ(1,K): = 0
For I = 1 to NIR
    For J = 1 to NIR
        XTV(I,J): = 0      XTXI(I,J): = 0
For J = 1 to NV
    For K = 1 to NV
        For I = 1 to NIR
            XTXI(J,K):=XTXI(J,K)+XMX(I,J)*XMX(I,K)
CALL INVERT(NV,IFLAG,XTXI)
For I = 1 to NIR
    For J = 1 to NIR
        VMAT(I,J): = VMATOR(I,J)
For J = 1 to NV
    For I = 1 to NIR
        For K = 1 to NIR
            XTV(J,I):=XTV(J,I)+XMX(K,J)*VMAT(K,I)
```

```
/        For I = 1 to NIR
/   /        For J = 1 to NIR
/   /        VMAT(I,J): = 0
/        For I = 1 to NV
/   /        For J = 1 to NV
/   /   /        For K = 1 to NIR
/   /   /   VMAT(I,J):=VMAT(I,J)+XTV(I,K)*XMX(K,J)
/        For I = 1 to NIR
/   /        For J = 1 to NIR
/   /        XTV(I,J): = 0
/        For I = 1 to NV
/   /        For J = 1 to NV
/   /   /        For K = 1 to NV
/   /   /    XTV(I,J):=XTV(I,J)+XTXI(I,K)*VMAT(K,J)
/        For I = 1 to NIR
/   /        For J = 1 to NIR
/   /        VMAT(I,J): = 0
/        For I = 1 to NV
/   /        For J = 1 to NV
/   /   /        For K = 1 to NV
/   /   /    VMAT(I,J):=VMAT(I,J)+XTV(I,K)*XTXI(K,J)
/        For I = 1 to NV
/        SDNLS(I): = SQRT(VMAT(I,I))
```

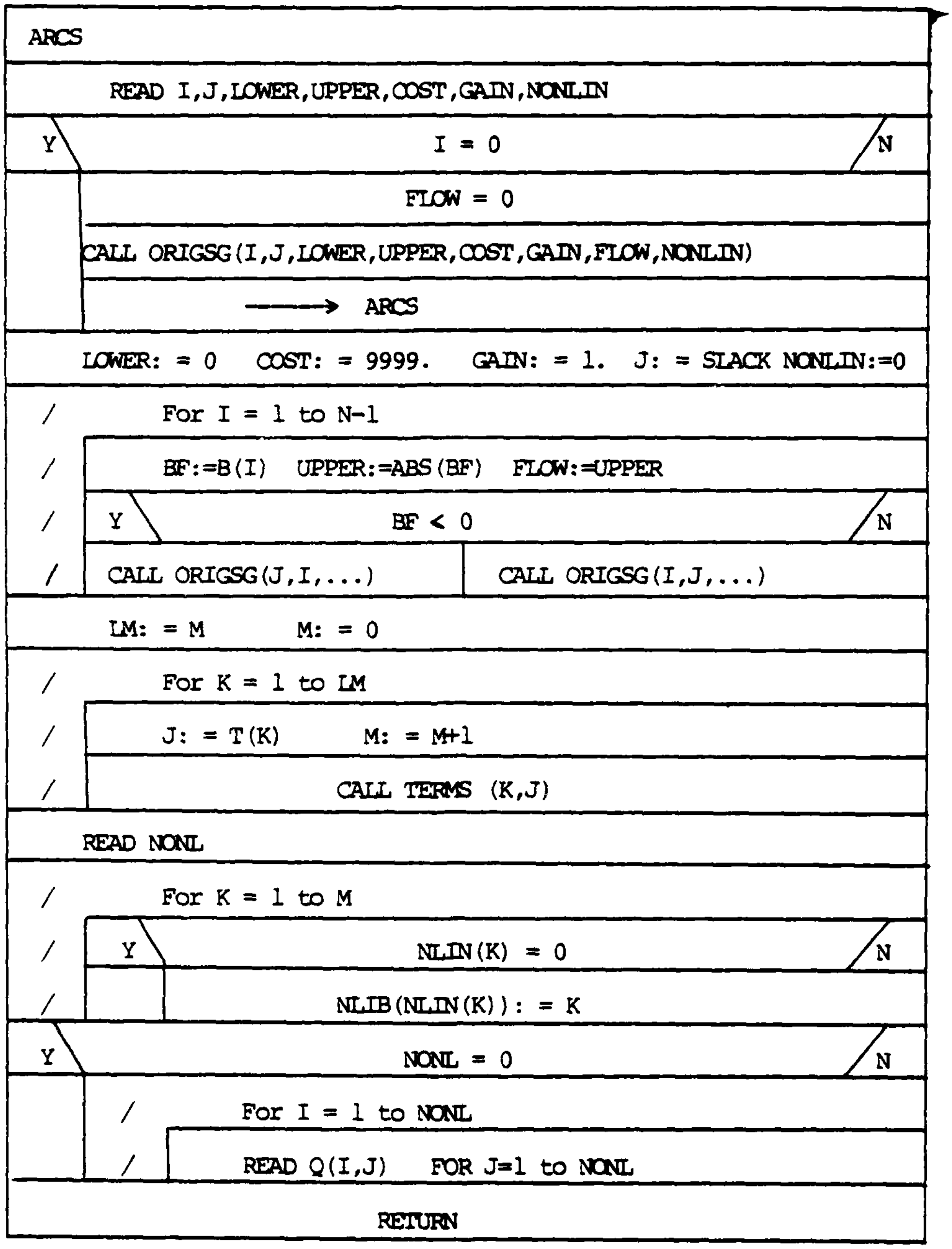
```
For I = 1 to NIR
    For J = 1 to NIR
        VMAT(I,J): = 0
For I = 1 to NV
    For J = 1 to NIR
        For K = 1 to NV
            VMAT(I,J):=VMAT(I,J)+XTXI(I,K)*XMX(J,K)
For I = 1 to NIR
    YVEC(I): = 0
For I = 1 to NV
    For J = 1 to NIR
        YVEC(I):=YVEC(I)+VMAT(I,J)*ZZT(J)
WZ: = YVEC(I)          NX: = NV -1
For I = 1 to NX
    QQ(1,I): = YVEC(I+1)
IT: = IT - 1
ILAST: = ILAST - 1
ILAST = 0
  Y:
    ILAST: = 12
  N:
For IK = 1 to IR
    HL(NGIB(IK)): = QQ(1,IK)
K: = IR + 1
```



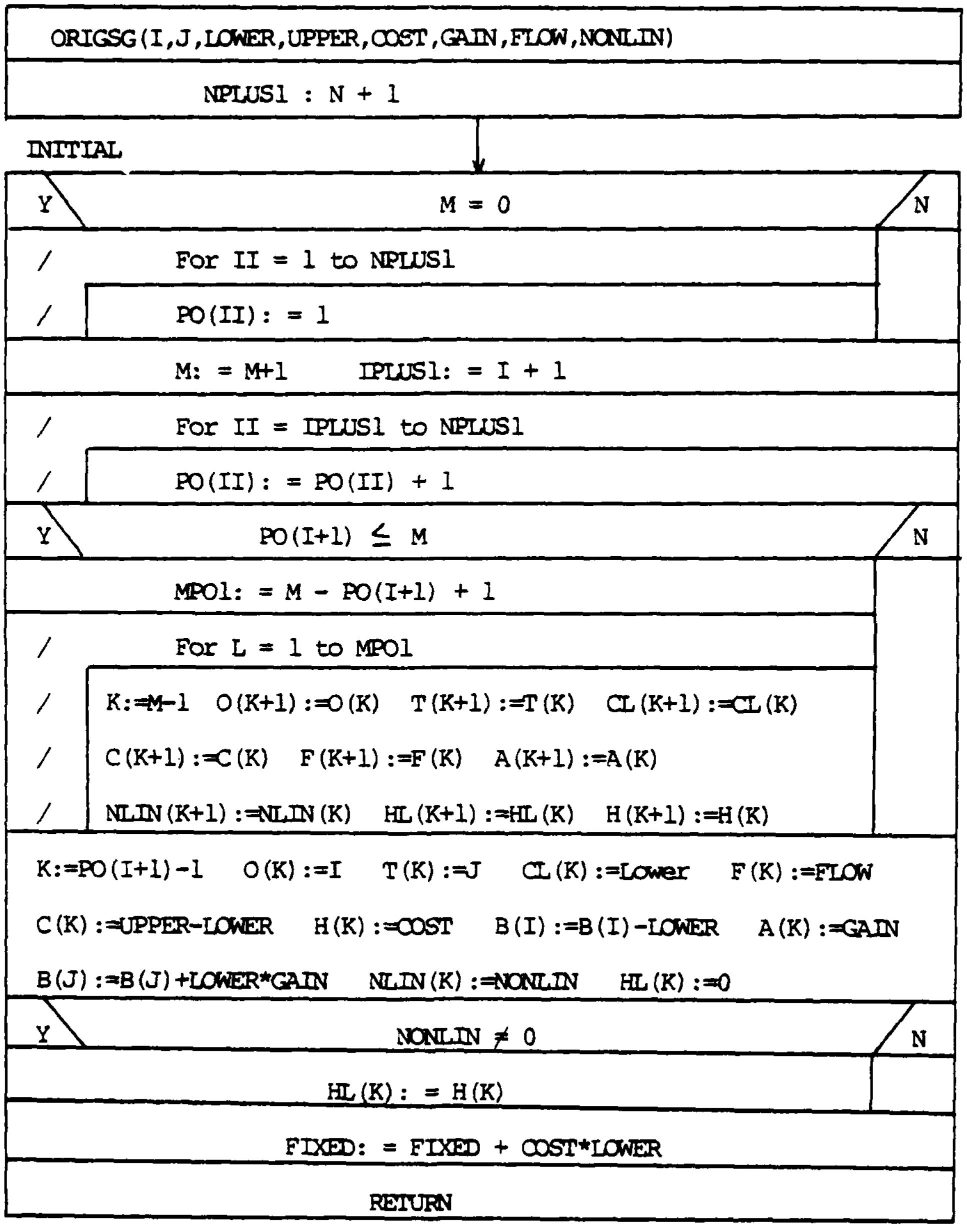
```
For I = 1 to IR
    Q(I,I): = QQ(1,K)
    K: = K + 1
IP: = IR - 1
For I = 1 to IP
    JJ: = I + 1
    For J = JJ to IR
        Q(I,J): = QQ(I,K)/2
        Q(J,I): = Q(I,J)
        K: = K + 1
IT = 0
Y
N
    ICURVE = 3
    Y
        —— PERIOD
    N
        READ ZMU,ZSIG
        ——> PERIOD
STOP
```



```
READG
READ N
M: = 0   SLACK: = N+1   N: = N+1
For I = 1 to N
    B(I): = 0
IR: = 0

READ
READ I,BF,BS,CS,IRESEV
I = 0
  Y:
    ——→ ARCS
  N:
B(I): = BF        IRES(I): = IRESEV
IRESEV = 0
  Y:
  N:
    IRESB(IRESEV): = I        AMT(IRESEV): = B(I)
IR: = IR+1
BS = 0
  Y:
    ——→ READ
  N:
BS > 0
  Y:
    J: = I   I: = SLACK
    LOWER: = 0   UPPER: = BS
    COST: = CS   GAIN: = 1
    FLOW: = 0   NONLIN: = 0
  N:
    J: = SLACK   LOWER: = 0
    UPPER: = -BS   COST: = CS
    GAIN: = 1   FLOW: = 0
    NONLIN: = 0
CALL ORIGSG(I,J,LOWER,UPPER,COST,GAIN,FLOW,NONLIN)
——→ READ
```

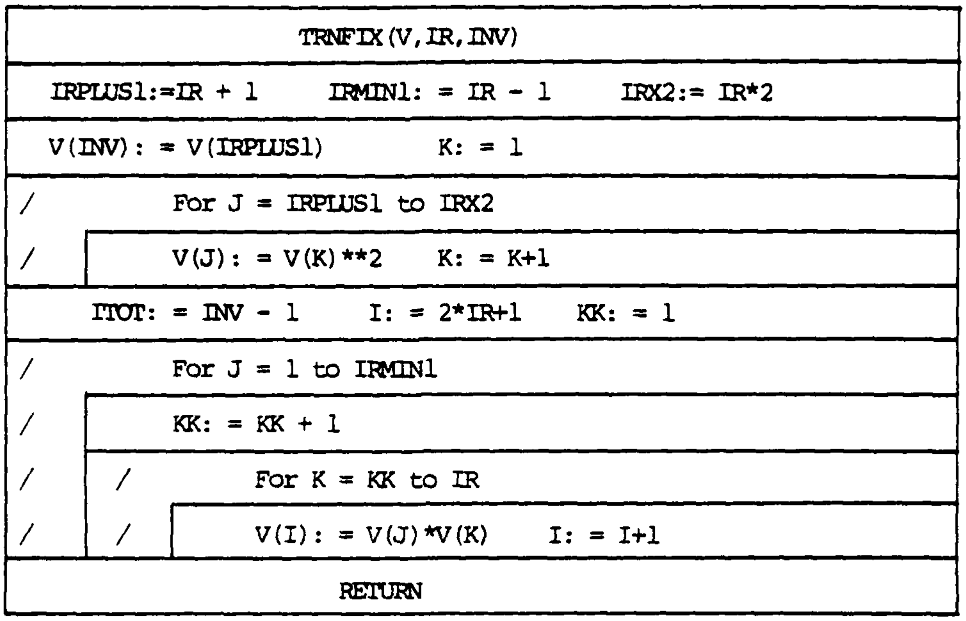


```
ARCS
  READ I,J,LOWER,UPPER,COST,GAIN,NONLIN
  Y  I = 0  N
    FLOW = 0
    CALL ORIGSG(I,J,LOWER,UPPER,COST,GAIN,FLOW,NONLIN)
    ———→ ARCS
  LOWER: = 0   COST: = 9999.   GAIN: = 1.   J: = SLACK NONLIN:=0
  For I = 1 to N-1
    BF:=B(I)   UPPER:=ABS(BF)   FLOW:=UPPER
    Y  BF < 0  N
      CALL ORIGSG(J,I,...)   |   CALL ORIGSG(I,J,...)
  LM: = M        M: = 0
  For K = 1 to LM
    J: = T(K)        M: = M+1
    CALL TERMS (K,J)
  READ NONL
  For K = 1 to M
    Y  NLIN(K) = 0  N
      NLIB(NLIN(K)): = K
  Y  NONL = 0  N
    For I = 1 to NONL
      READ Q(I,J)    FOR J=1 to NONL
  RETURN
```

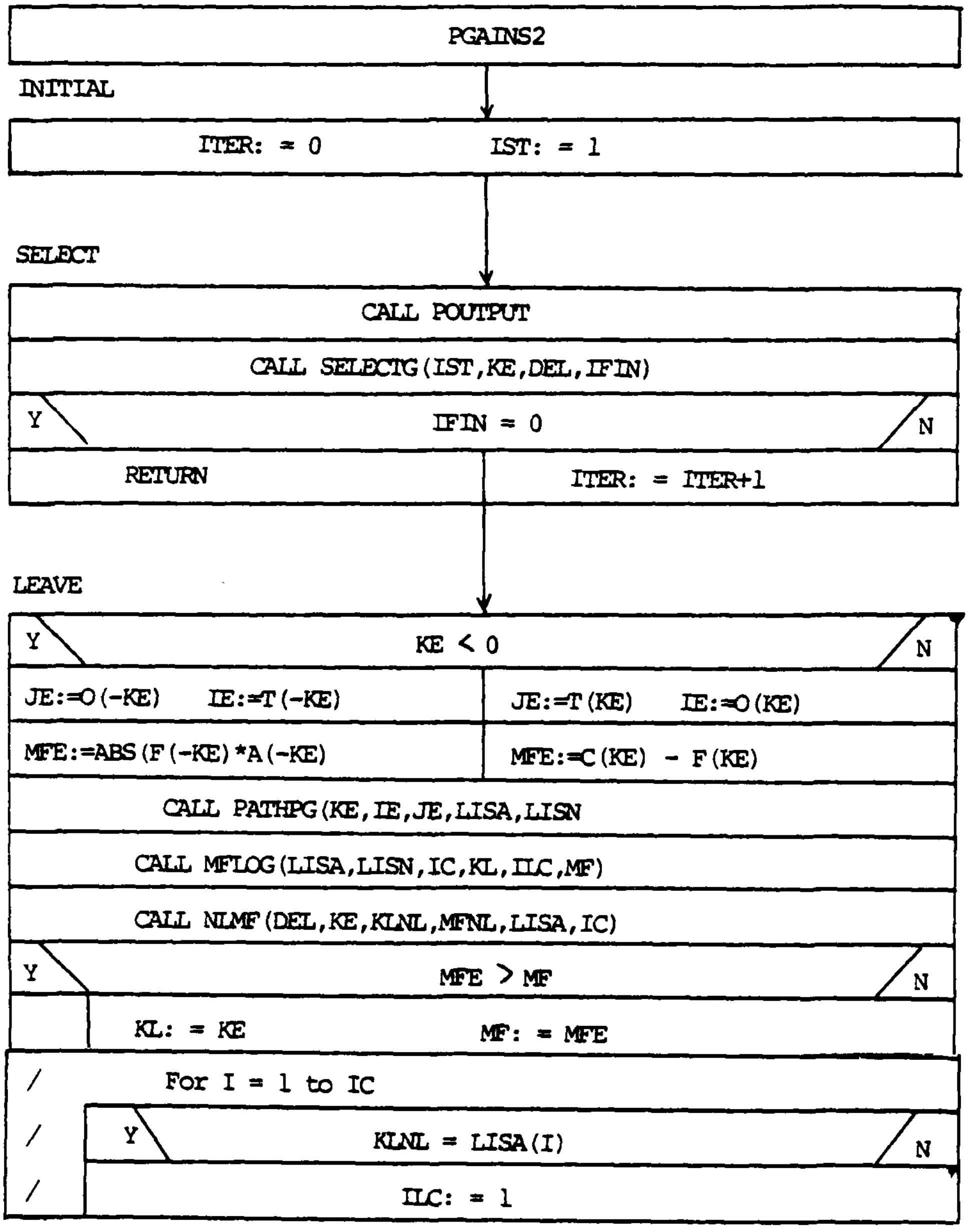


```
ORIGSG(I,J,LOWER,UPPER,COST,GAIN,FLOW,NONLIN)
        NPLUS1 : N + 1
INITIAL
Y                       M = 0                       N
    For II = 1 to NPLUS1
        PO(II): = 1
        M: = M+1      IPLUS1: = I + 1
    For II = IPLUS1 to NPLUS1
        PO(II): = PO(II) + 1
Y               PO(I+1) ≤ M                         N
        MPO1: = M - PO(I+1) + 1
    For L = 1 to MPO1
        K:=M-1  O(K+1):=O(K)  T(K+1):=T(K)  CL(K+1):=CL(K)
        C(K+1):=C(K)  F(K+1):=F(K)  A(K+1):=A(K)
        NLIN(K+1):=NLIN(K)  HL(K+1):=HL(K)  H(K+1):=H(K)
K:=PO(I+1)-1  O(K):=I  T(K):=J  CL(K):=Lower  F(K):=FLOW
C(K):=UPPER-LOWER  H(K):=COST  B(I):=B(I)-LOWER  A(K):=GAIN
B(J):=B(J)+LOWER*GAIN  NLIN(K):=NONLIN  HL(K):=0
Y                     NONLIN ≠ 0                    N
                HL(K): = H(K)
              FIXED: = FIXED + COST*LOWER
                      RETURN
```

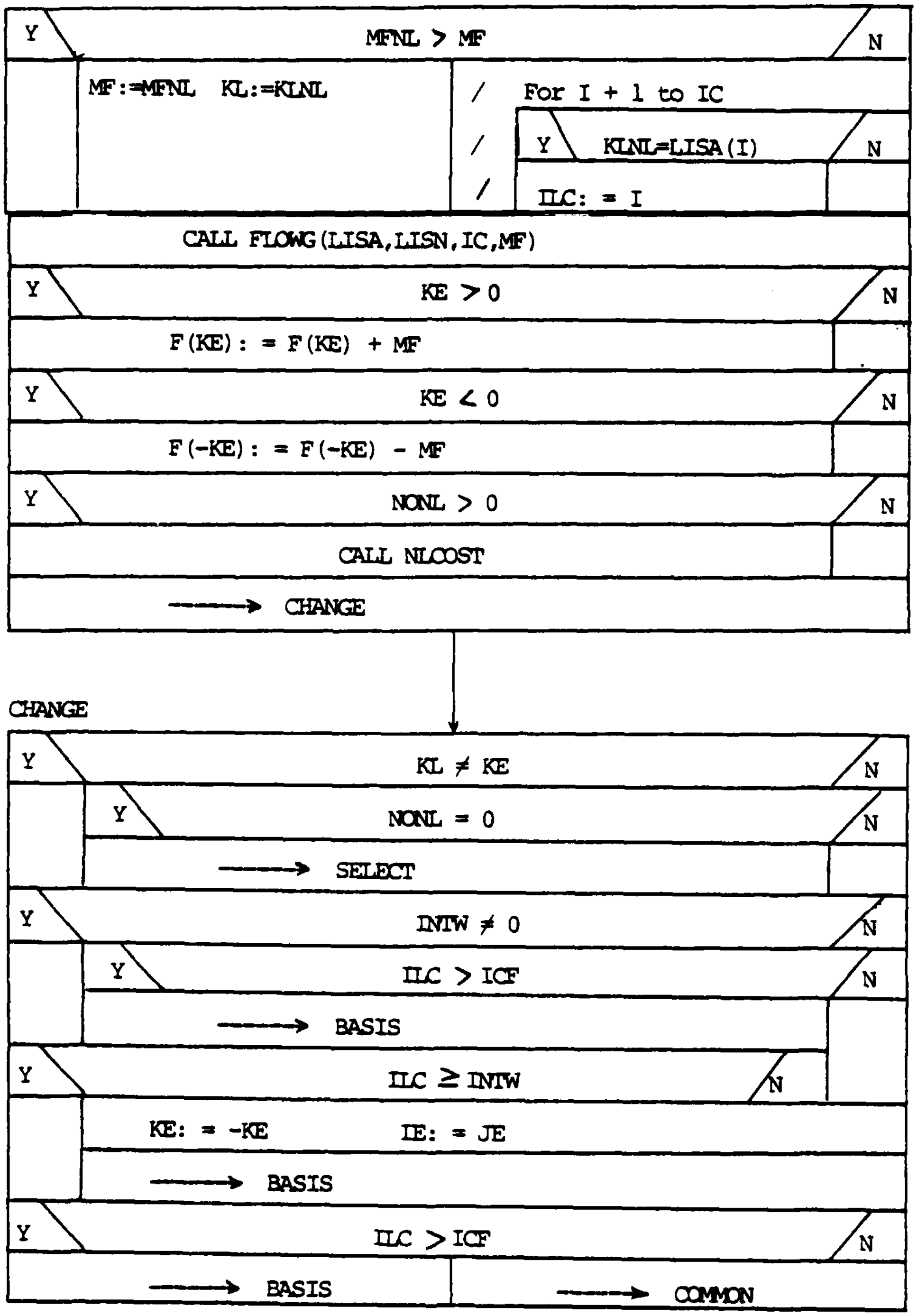
```
FIXINV(IR)
NOB: = NIR          NM: = IR + 1
/  For I = 1 to NV
/     V(I): = 0
/  For IK = 1 to NIR
/     I: = 1
/     /  For J = 1 to IR
/     /     V(J): = S(IK,J)
/     V(NM): = ZZT(IK)
/     INV: = NV
/     CALL TRNFIX(V,IR,INV)
/     /  For K = 1 to NV - 1
/     /     XMX(IK,K+1): = V(K)
/     XMX(IK,1): = 1
RETURN
```



```
TRNFIX(V,IR,INV)
IRPLUS1:=IR + 1     IRMIN1: = IR - 1     IRX2:= IR*2
V(INV): = V(IRPLUS1)     K: = 1
For J = IRPLUS1 to IRX2
    V(J): = V(K)**2     K: = K+1
ITOT: = INV - 1     I: = 2*IR+1     KK: = 1
For J = 1 to IRMIN1
    KK: = KK + 1
    For K = KK to IR
        V(I): = V(J)*V(K)     I: = I+1
RETURN
```



```
PGAINS2
INITIAL
    ITER: = 0        IST: = 1
SELECT
    CALL POUTPUT
    CALL SELECTG(IST,KE,DEL,IFIN)
    IFIN = 0
      Y: RETURN
      N: ITER: = ITER+1
LEAVE
    KE < 0
      Y: JE:=O(-KE)     IE:=T(-KE)
         MFE:=ABS(F(-KE)*A(-KE)
      N: JE:=T(KE)      IE:=O(KE)
         MFE:=C(KE) - F(KE)
    CALL PATHPG(KE,IE,JE,LISA,LISN
    CALL MFLOG(LISA,LISN,IC,KL,ILC,MF)
    CALL NLMF(DEL,KE,KLNL,MFNL,LISA,IC)
    MFE > MF
      Y:
      N: KL: = KE       MF: = MFE
    For I = 1 to IC
        KLNL = LISA(I)
          Y:
          N: ILC: = 1
```


Y
MFNL > MF
N
MF:=MFNL KL:=KLNL
For I + 1 to IC
Y
KLNL=LISA(I)
N
ILC: = I
CALL FLOWG(LISA,LISN,IC,MF)
Y
KE > 0
N
F(KE): = F(KE) + MF
Y
KE ∠ 0
N
F(-KE): = F(-KE) - MF
Y
NONL > 0
N
CALL NLCOST
→ CHANGE
CHANGE
Y
KL ≠ KE
N
Y
NONL = 0
N
→ SELECT
Y
INTW ≠ 0
N
Y
ILC > ICF
N
→ BASIS
Y
ILC ≥ INTW
N
KE: = -KE
IE: = JE
→ BASIS
Y
ILC > ICF
N
→ BASIS
→ COMMON

COMMON

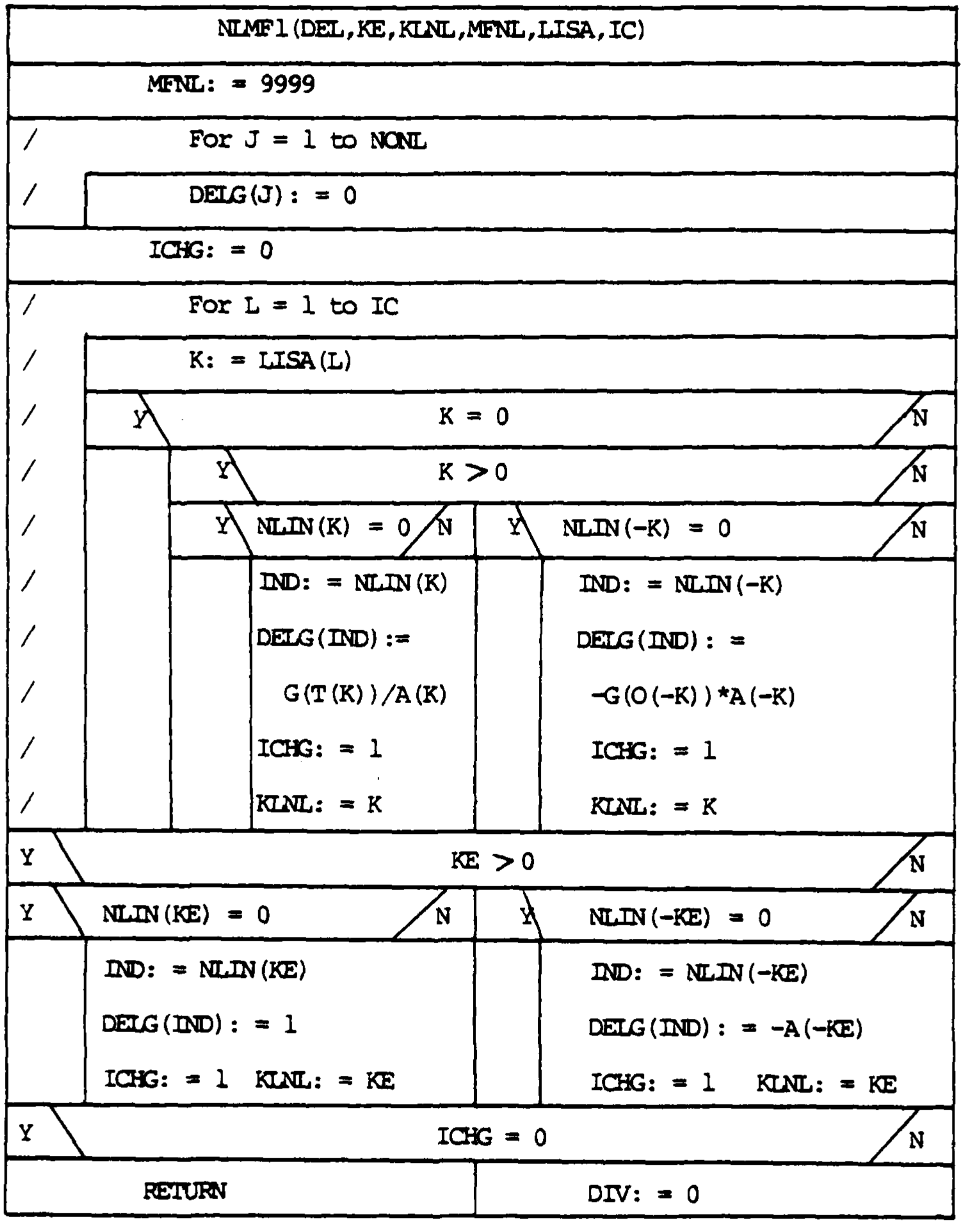
```
KE = KL
  Y
  N
    -KE = KL
      Y
      N
        CALL TRECHGW(KL,KE)
NONL = 0
  Y
  N
    CALL NLARCS
    IINOL = 0
      Y
        CALL CYCLE(IE,BET,COST)
        PI(IE): = COST/(BET-1)
      N
        CALL NLPI(KE)
        ———→ SELECT
———→ POTENTIAL
```

BASIS

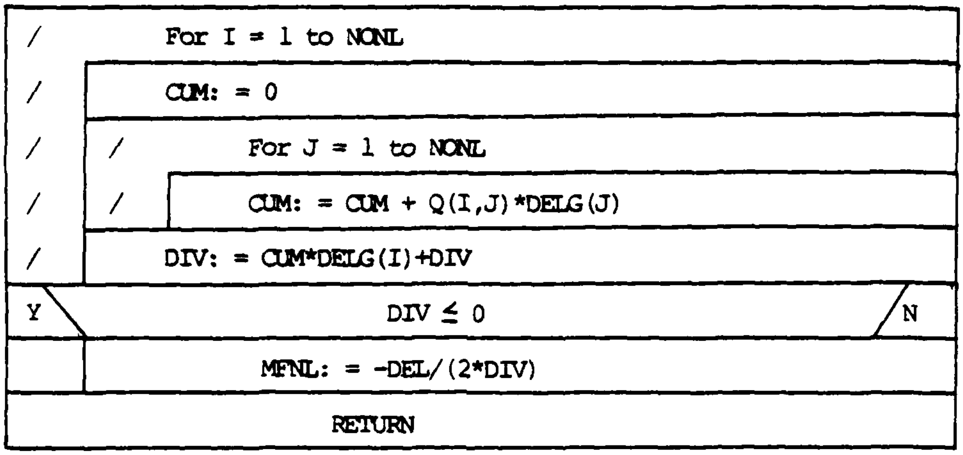
```
KE = KL
  Y
  N
    -KE = KL
      Y
      N
        CALL TRECHGW(KL,KE)
NONL = 0
  Y
    ———→ POTENTIAL
  N
    CALL NLARCS
IINOL ≠ 0
  Y
    CALL NLPI(KE)
    ———→ SELECT
  N
    ———→ POTENTIAL
```

POTENTIAL

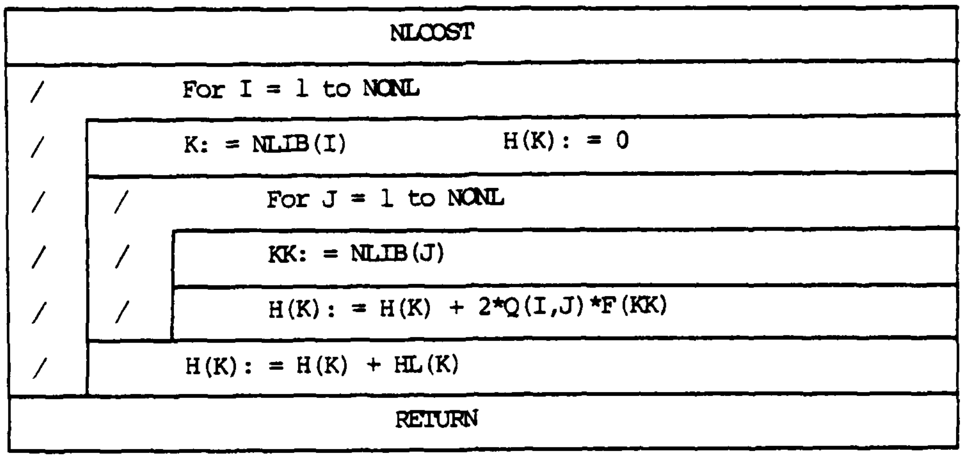
```
CALL DUAL(IE)
——— SELECT
```



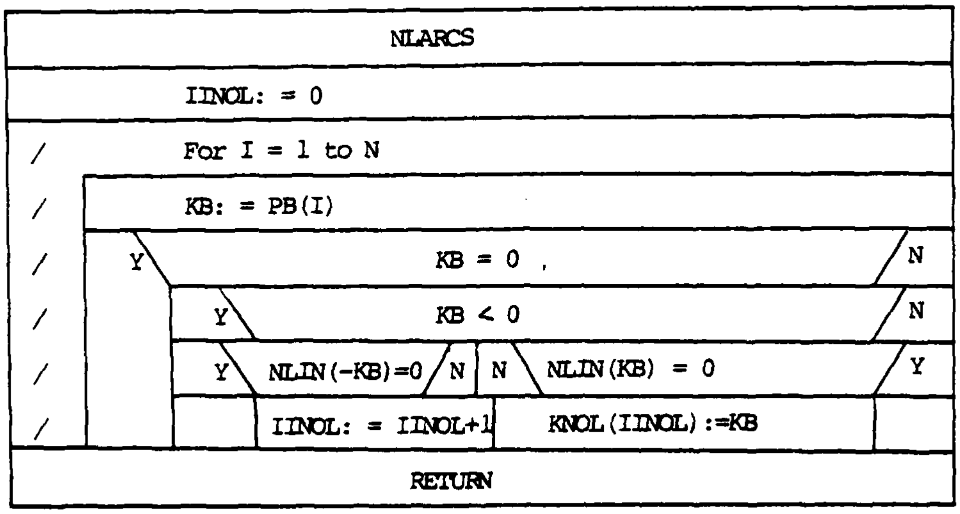
```
NLMF1(DEL,KE,KLNL,MFNL,LISA,IC)
MFNL: = 9999
For J = 1 to NONL
    DELG(J): = 0
ICHG: = 0
For L = 1 to IC
    K: = LISA(L)
    K = 0
      Y:
      N:  K > 0
            Y:  NLIN(K) = 0
                  Y:
                  N:  IND: = NLIN(K)
                      DELG(IND):=
                        G(T(K))/A(K)
                      ICHG: = 1
                      KLNL: = K
            N:  NLIN(-K) = 0
                  Y:
                  N:  IND: = NLIN(-K)
                      DELG(IND): =
                        -G(O(-K))*A(-K)
                      ICHG: = 1
                      KLNL: = K
KE > 0
  Y:  NLIN(KE) = 0
        Y:
        N:  IND: = NLIN(KE)
            DELG(IND): = 1
            ICHG: = 1  KLNL: = KE
  N:  NLIN(-KE) = 0
        Y:
        N:  IND: = NLIN(-KE)
            DELG(IND): = -A(-KE)
            ICHG: = 1  KLNL: = KE
ICHG = 0
  Y:  RETURN
  N:  DIV: = 0
```



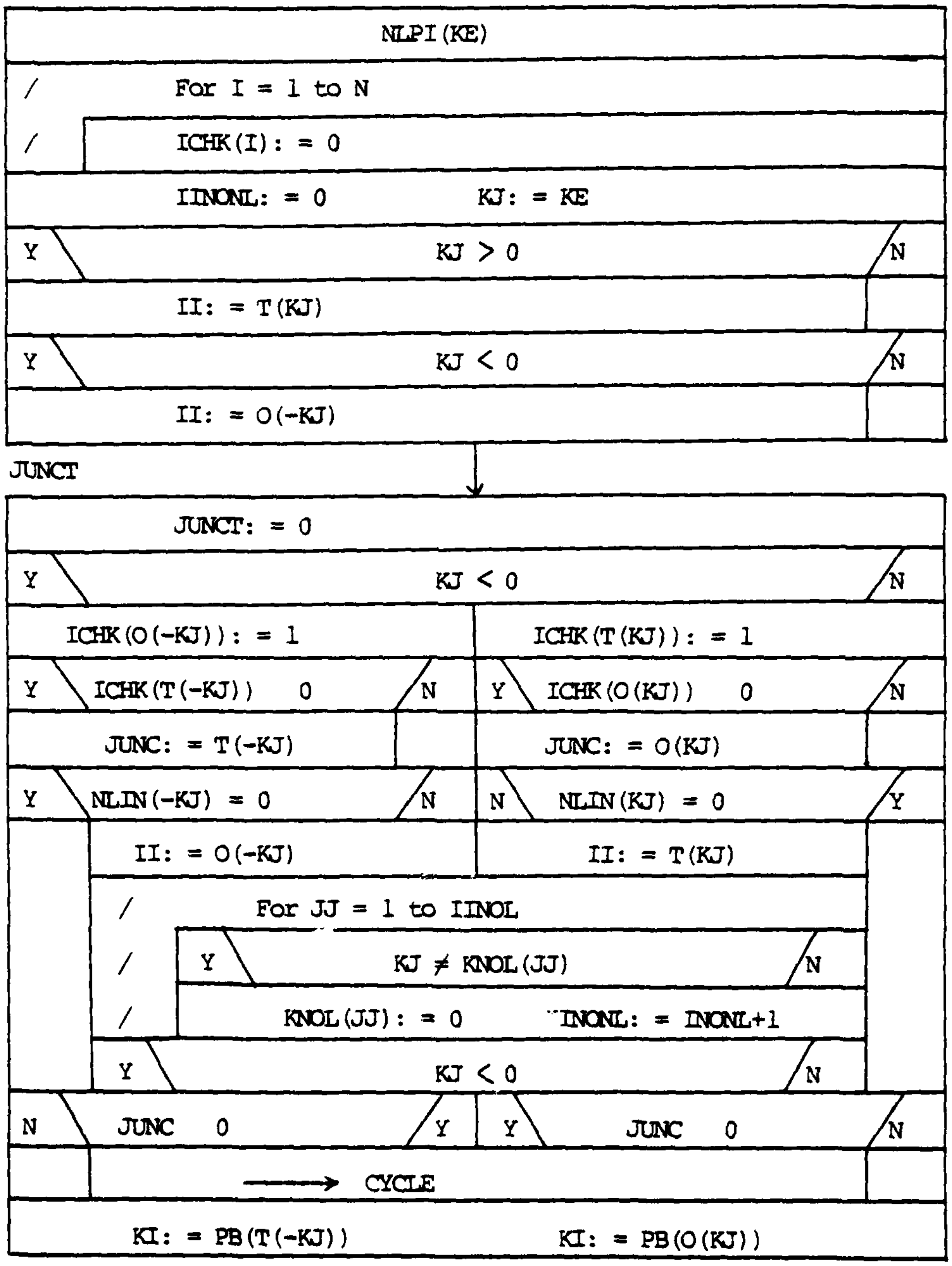
```
For I = 1 to NONL
    CUM: = 0
    For J = 1 to NONL
        CUM: = CUM + Q(I,J)*DELG(J)
    DIV: = CUM*DELG(I)+DIV
DIV ≤ 0
    Y
    N
        MFNL: = -DEL/(2*DIV)
RETURN
```



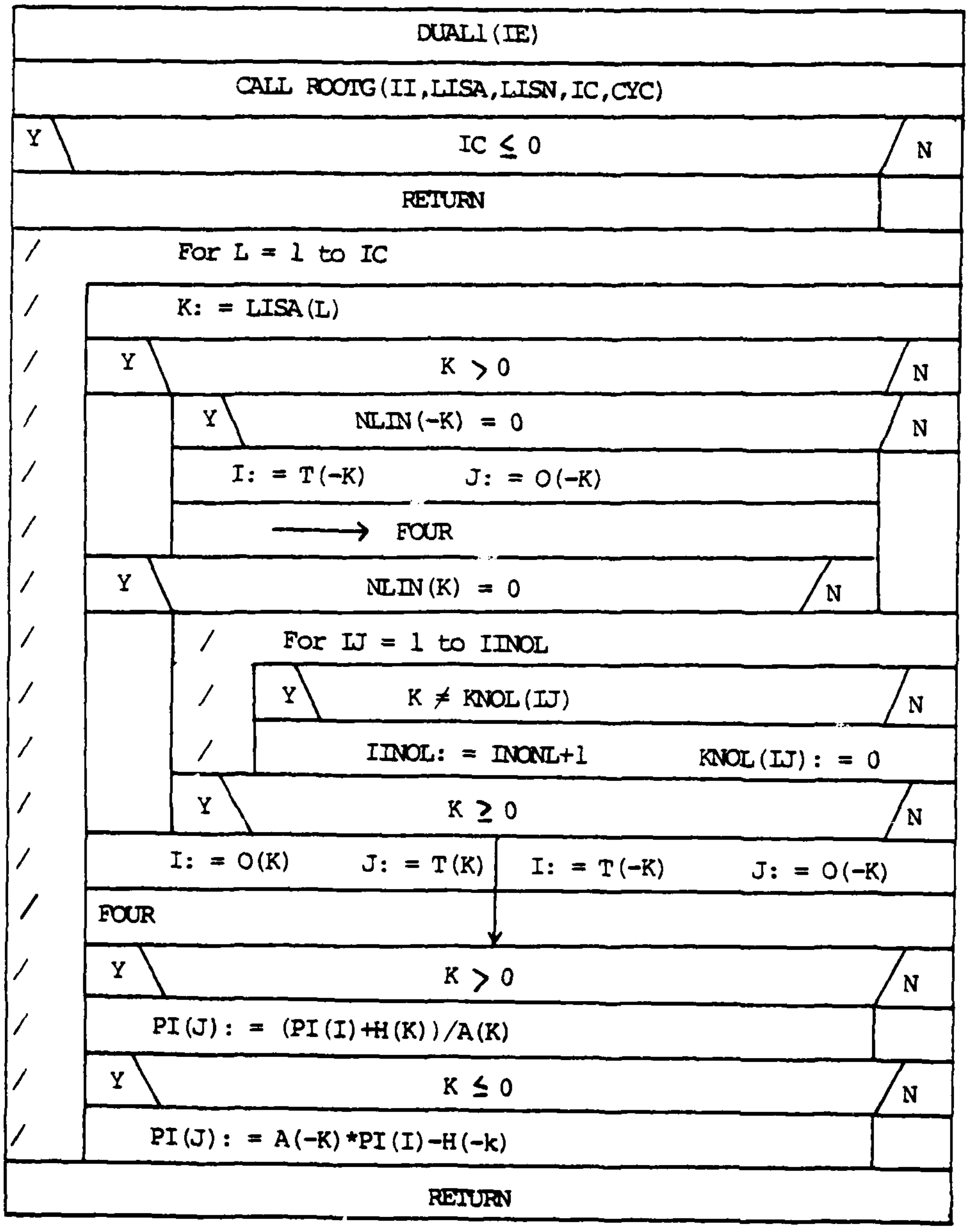
```
NLCOST
For I = 1 to NONL
    K: = NLIB(I)          H(K): = 0
    For J = 1 to NONL
        KK: = NLIB(J)
        H(K): = H(K) + 2*Q(I,J)*F(KK)
    H(K): = H(K) + HL(K)
RETURN
```



```
NLARCS
IINOL: = 0
For I = 1 to N
    KB: = PB(I)
    Y        KB = 0 ,        N
        Y      KB < 0        N
            Y NLIN(-KB)=0 N | N NLIN(KB) = 0 Y
                IINOL: = IINOL+1     KNOL(IINOL):=KB
RETURN
```


NLPI(KE)
/
For I = 1 to N
/
ICHK(I): = 0
IINONL: = 0
KJ: = KE
Y
KJ > 0
N
II: = T(KJ)
Y
KJ < 0
N
II: = O(-KJ)
JUNCT
JUNCT: = 0
Y
KJ < 0
N
ICHK(O(-KJ)): = 1
ICHK(T(KJ)): = 1
Y
ICHK(T(-KJ)) 0
N
Y
ICHK(O(KJ)) 0
N
JUNC: = T(-KJ)
JUNC: = O(KJ)
Y
NLIN(-KJ) = 0
N
N
NLIN(KJ) = 0
Y
II: = O(-KJ)
II: = T(KJ)
/
For JJ = 1 to IINOL
/
Y
KJ ≠ KNOL(JJ)
N
/
KNOL(JJ): = 0
INONL: = INONL+1
Y
KJ < 0
N
N
JUNC 0
Y
Y
JUNC 0
N
CYCLE
KI: = PB(T(-KJ))
KI: = PB(O(KJ))

```
Y                    KI = 0                         N
        | KJ: = KI
        | ———→ JUNCT
KK: = PB(II)
Y                    KK = 0                         N
        | Y               KK < 0                    N
        | I: = T(-KK)                   | PI(II): = PI(O(KK)) +
        | PI(II):=A(-KK)*PI(I)-H(-KK)   | H(KK))/A(KK)
        | ———→ DUAL1
Y                    KE > 0                         N
II: = O(KE)                 | II: = T(-KE)
PI(II): =0
———→ DUAL1

CYCLE
CALL CYCLE(JUNC,BET,COST)
II: = JUNC          PI(II): = COST/(BET-1)

DUAL1
CALL DUAL1
Y                INONL = IINOL                      N
RETURN                      | IJ: = 1

SEARCH
Y                KNOL(IJ) ≠ 0                       N
KJ: = KNOL(IJ)              | IJ: = IJ+1
/   For J = 1 to N          |
/     ICHK(J): = 0          | ———→ SEARCH
———→ Junct
```



```
DUAL1(IE)
CALL ROOTG(II,LISA,LISN,IC,CYC)
Y  IC ≤ 0  N
    RETURN
For L = 1 to IC
    K: = LISA(L)
    Y  K > 0  N
        Y  NLIN(-K) = 0  N
            I: = T(-K)        J: = O(-K)
            ——→ FOUR
    Y  NLIN(K) = 0  N
        For LJ = 1 to IINOL
            Y  K ≠ KNOL(LJ)  N
                IINOL: = INONL+1        KNOL(LJ): = 0
        Y  K ≥ 0  N
    I: = O(K)        J: = T(K)   |   I: = T(-K)        J: = O(-K)
    FOUR
    Y  K > 0  N
        PI(J): = (PI(I)+H(K))/A(K)
    Y  K ≤ 0  N
        PI(J): = A(-K)*PI(I)-H(-k)
RETURN
```

Ali, A.I., Helgason, R.V. and Kennington, J.L., "The Convex Cost Network Flow Problem: A State-of-the Art Survey", Technical Report OREM 78001, Southern Methodist University, January 1978.

Anderson, V.L. and McLean, R.A., Design of Experiments: A Realistic Approach, Marcel Dekker, New York, 1974.

Askew, A.J., "Chance-Constrained Dynamic Programming and the Optimization of Water Resources Systems", Water Resources Research, Vol. 10, p. 1099-1106, 1974.

Barnes, J.W., Zinn, C.D. and Eldred, B.S., "A Methodology for Obtaining the Probability Density Function of the Present Worth of Probabilistic Cash Flow Profiles", AIIE Transactions, Vol. 10, No. 3, p. 226-235, 1978.

Becker, L. and Yeh, W.W.G., "Optimization of Real Time Operation of a Multiple-Reservoir System", Water Resources Research, Vol. 10, p. 1107-1112, 1974.

Bellman, R. and Dreyfus, S., Applied Dynamic Programming, Princeton University Press, 1962.

Bhaumik,G. Optimum Operating Policies Of A Water Distribution System With Losses, Unpublished Dissertation, The University of Texas at Austin, 1973.

Bodin, L.D. and Roefs, T.G., "A Decomposition Approach to Nonlinear Programs as Applied to Reservoir Systems", Networks, Vol. 1, p. 59-73, 1971.

Bowker, A.H. and Lieberman, G.J., Engineering Statistics, Prentice Hall, Englewood Cliffs, New Jersey, 1972.

Box, G.E.P. and Lucas, H.L., "Design of Experiments in Nonlinear Situations", Biometrica 46, p. 77-90, 1959.

Box, M.J. and Draper, N.R., "Factorial Designs, The X'X Criterion, and Some Related Matters", Technometrics, Vol. 13, No. 4, p. 731-743, 1971.

Bradley, S.P., Hax, A.C. and Magnanti, T.L., Applied Mathematical Programming, Addison Wesley, Reading, Mass., 1977.

Buras, Nathan, Scientific Allocation Of Water Resources, American Elsevier Co., New York, N.Y., 1972.

Burr, I.W., Applied Statistical Methods, Academic Press, New York, 1974.

Butcher, William S., "Stochastic Dynamic Programming For Optimal Reservoir Operation", Water Resources Bulletin, Vol. 7, No. 1, February 1971, p. 115-123.

Butcher, William S., "Stochastic Dynamic Programming and Water Resources Management", Presented to Operations Research Society of America in San Diego, California, November 14, 1973.

Chu, H.W., Stochastic Network With Recourse, Ph.D. Dissertation, The University of Texas at Austin, 1980.

Clark, T.C. and Schkade, L.L., Statistical Analysis for Administrative Decisions, South Western Publishing Co., Cincinnati, Ohio, 1974.

Cooper, L. and Kennington, J.L., "Steady State Analysis of Nonlinear Resistive Electrical Networks Using Optimization Techniques", Technical Report, IEOR 77012, Southern Methodist University, October, 1977.

Croley, T.E., "Sequential Deterministic Optimization in Reservoir Operation", Journal Hydraulics Division, A.S.C.E., Vol. 100, HY3, p. 443-459, 1974b.

Croley, T.E., "Sequential Stochastic Optimization for Reservoir System", Journal Hydraulics Division, A.S.C.E., Vol. 100, HY1, p. 201-219, 1974a.

Curry, Guy L. and Helm, James C., "A Chance-Constrained Model For A Single Multipurpose Reservoir System", Texas A&M University, 1972.

Curry, Guy L., Helm, James C., and Clark, Robert A., "A Chance-Constrained Programming Model For A Corrected System Of Multipurpose Reservoirs", Texas A&M University, 1972.

Daniel, C. and Wood, F.S., Fitting Equations to Data, Wiley-Interscience, New York, N.Y., 1971.

Dembo, R.S. and Klincewicz, J.G., "A Scaled Reduced Gradient Algorithm for Network Flow Problems with Convex Separable Costs", Working Paper Series B, Yale School of Organization and Management, November, 1979.

Doran, D.G., "An Efficient Transition For Discrete State Reservoir Analysis: The Divided Interval Technique", Water Resources Research, Vol. 11, No. 6, December 1975, p. 867-873.

Draper, N.R. and Smith, H., Applied Regression Analysis, Wiley, New York, N.Y., 1969.

Driscoll, W.D., Multireservoir Operating Policies Considering Uncertainty, Unpublished Dissertation, University of Texas at Austin, September 1974.

Drobny, N.L., "Linear Programming Application in Water Resources", Water Resources Bulletin, Vol. 7, p. 1180-1193, 1971.

Erickson, L.E., "A Nonlinear Model of a Water Reservoir System with Multiple Uses and Its Optimization by Combined Use of Dynamic Programming and Pattern Search Techniques", Water Resources Bulletin, Vol. 5, p. 18-36, 1969.

Evenson, D.E. and Mosely, J.C., "Simulation/Optimization Techniques For Multibasin Water Resources Planning", Water Resources Bulletin, Vol. 6, No. 5, p. 725-736, 1975.

Florian, M., "An Improved Linear Approximation Algorithm for the Network Equilibrium (Packet Switching) Problem", Publication 3251, Department d'informatique et de recherche operationelle, Universite de Montreal, 1977.

Frank, M. and Wolfe, P., "An Algorithm for Quadratic Programming", Naval Research Logistics Quarterly 3, 1956.

Fults, D.M., Hancock, L. and Logan, G., "A Practical Monthly Optimum Operations Model", Journal of the Water Resources Planning and Management Division, Proceedings of A.S.C.E., Vol. 102, No. WR1, April 1976.

Fults, D.M. and Hancock, L.F., "Optimum Operations Model for Shasta-Trinity System", Journal of Hydraulics Division, A.S.C.E., Vol. 98, No. HY9, p. 1497-1514, 1972.

Gagnon, C.R., Hicks, R.H., Jacoby, S.L.S. and Kowalik, J.S., "A Nonlinear Programming Approach to a Very Large Hydroelectric System Optimization", Mathematical Programming, Vol. 6, p. 28-41, 1974.

Gundelach, J. and ReVelle, C., "Linear Decision Rule in Reservoir Management and Design, 5. A General Algorithm", Water Resources Research, Vol. 11, No. 2, April 1975, p. 204,207.

Hall, W.A., Butcher, W.S. and Esogbue, A., "Optimization of the Operation of a Multiple-Purpose Reservoir by Dynamic Programming", Water Resources Research, Vol. 4, p. 471-477, 1968.

Hall, Warren A., and Dracup, John A., Water Resources Systems Engineering, McGraw Hill Book Co., New York, N.Y., 1970.

Heidari, M., Chow, V.T., Kokotovic, P.V. and Meredith, D.D., "Discrete Differential Dynamic Programming Approach to Water Resources Systems Optimization", Water Resources Research, Vol. 7, p. 273-282, 1971.

Helgason, R.V. and Kennington, J.L., An Efficient Specialization of the Convex Simplex Method for Nonlinear Network Flow Problems", Technical Report, IEOR 77017, Southern Methodist University, April, 1978.

Helm, James C., Curry, Guy L., and Hasan, Sayeed, "A Capacity Expansion Model For A System Of Linked Multi-purpose Reservoirs With Stochastic Inflows", Texas A&M University, 1972.

Hicks, C.R., Fundamental Concepts in the Design of Experiments, Holt, Rinehart and Winston, Inc., New York, N.Y., 1973.

Hirsch, R.M., Cohon, J.L., and ReVelle, C.S., "Gains From Joint Operation of Multiple Reservoir Systems", Water Resources Research, Vol. 13, No. 2, April 1977, p. 239-245.

Houck, M.H. and Cohon, J.L., "Sequentially Explicitly Stochastic Linear Programming Models: A Proposed Method for Design and Management of Multipurpose Reservoir Systems", Water Resources Research, Vol. 14, No. 2, April 1978, p. 161-169.

Jensen, P.A. and Barnes, J.W., Network Flow Programming, Wiley and Sons, New York, N.Y., June 1980.

Jensen, P.A., Bhaumik, G. and Driscoll, W., "Network Flow Modeling of Multireservoir Distribution Systems", Center for Research in Water Resources, University of Texas at Austin, 1974.

Jensen, P.A., Chu, H.W. and Cochard, D.D., "Network Flow Optimization For Water Resources Planning With Uncertainties in Supply and Demand", Center for Research in Water Resources Bureau of Engineering Research, Technical Report CRWR-172, University of Texas at Austin, July 1980.

Jensen, P.A. and Reeder, Hugh A., "Solution of Convex Network Problems With A Revised Out-Of-Kilter Algorithm", University of Texas, 1974.

Joeres, E.F., Liebman, J.C. and ReVelle, C.S., "Operating Rules for Joint Operation of Raw Water Sources", Water Resources Research, Vol. 7, p. 225-235, 1971.

Kennington, J.L. and Helgason, R.B., Algorithms for Network Programming, Wiley Interscience, 1980.

Kerr, J.A., "Multireservoir Analysis Techniques in Water Quantity Studies", Water Resources Bulletin, Vol. 8, p. 871-880, 1972.

Kiefer, J., "Optimum Designs in Regression Problems, II", Ann. Math Statist. 32, p. 298 325, 1961.

Klemes, V., "Discrete Representation Of Storage For Stochastic Reservoir Optimization", Water Resources Research, Vol. 13, No. 1, February 1977, p. 149-158.

Klincewicz, J.G., "Algorithms for Network Flow Problems with Convex Separable Costs", Ph.D. Dissertation, Yale University, New Haven, 1979.

Lane, Morton, "Conditional Chance-Constrained Model for Reservoir Control", Water Resources Research, Vol. 9, No. 4, August 1973.

Lasdon, L.S., "A Survey of Users of Nonlinear Programming", presented at ORSA/TIMS Joint National Meeting, Miami, 1976.

Lee, E.S. and Waziruddin, S., "Applying the Gradient Projection and Conjugate Gradient to the Optimum Operation of Reservoirs", Water Resources Bulletin, Vol. 6, p. 713-724, 1970.

Linsley, R.K. and Franzini, J.B., Water-Resources Engineering, McGraw-Hill, Inc., New York, 1964.

Liu, C.S. and Tedrow, A.C., "Multilake River System Operation Rules", Journal of Hydraulics Division, A.S.C.E., Vol. 99, No. HY9, p. 1369-1381, 1973.

Loucks, Daniel P., "Stochastic Methods for Analyzing River Basin Systems", Cornell University Water Resources Center, 1969, Technical Report, No. 16.

Loucks, Daniel P., "Computer Model for Reservoir Regulation", Proceedings, American Society of Civil Engineers, 94(SA4), 1968, p. 657-669.

Loucks, D.P. and Dorfman, P.J., "An evaluation of Some Linear Decision Rules In Chance-Constrained Models for Reservoir Planning and Operation", Water Resources Research, Vol. 11, No. 6, December 1969, p. 767-777.

Loucks, D.P. and Falkson, L.M., "Comparison of Some Dynamic, Linear and Policy Iteration Methods for Reservoir Operation", Water Resources Bulletin, vol. 6, p. 384-400, 1970.

Luenberger, David G., Introduction to Linear and Nonlinear Programming, Addison-Wesley, Reading, Mass., 1965.

Mannos, R., "Application of Linear Programming to Efficiency in Operation for a System of Dams", Econometrica, Vol. 33, p.335-336, 1955.

Mawer, P.A. and Thorn, D., "Improved Dynamic Programming Procedures and Their Practical Application to Water Resource Systems", Water Resources Research, Vol. 10, p. 183-190, 1974.

Meier, W.L. and Beightler, C.S., "An Optimization Method for Branching Multistage Water Resource Systems", Water Resources Research, Vol. 3, p. 645-652, 1967.

Mejia, J.M., Egli, P. and LeClerc, A., "Evaluating Multireservoir Operating Rules", Water Resources Research, Vol. 10, p. 1090-1098, 1974.

Meredith, D.D., "Optimal Operation of Multiple Reservoir System", Journal Hydraulic Division, A.S.C.E., Vol. 101, No. HY2, p. 299-312, 1975.

Minieka, E., Optimization Algorithms for Networks and Graphs, Marcel Dekker, Inc., New York, 1978.

Mitchell, T.J., "An Algorithm for the Construction of "D-Optimal" Experimental Designs", Technometrics, Vol. 16, No. 2, p. 203-210, May 1974a.

Mitchell, T.J., "Computer Construction of "D-Optimal" First Order Designs", Technometrics, Vol. 16, No. 2, p. 211-220, May 1974b.

Moran, P.A.P., "A Probability Theory of Dams and Storage Systems", Aust. Journal of Applied Science, Vol. 5, p. 116-124, 1954.

Parikh, Shailendra C., "Linear Dynamic Decomposition Multipurpose Reservoir System", Operations Research Center, University of California, Berkeley, California, September 1966.

Parikh, Shailendra C. and Shepherd, R.W., "Linear Dynamic Decomposition Programming Approach to Long Range Optimization of Northern California Water Resources System", Operations Research Center Report 67-30, University of California, Berkeley, California, August 1967.

Pinter, J., "A Stochastic Programming Model Applied to Water Resources Management", Computing Center for Universities, Budapest, Hungary, Abstract in Selected Water Resources Abstracts, Vol. 9, No. 21, p. 24, 1976.

Prekopa, A., Rapcsak, T. and Zsuffa, I., "Serially Linked Reservoir Design Using Stochastic Programming", Water Resources Research, Vol. 14, No. 4, August 1968, p. 672-678.

ReVelle, C. and Gundelach, J., "Linear Decision Rule in Reservoir Management and Design, 4. A Rule That Minimizes the Output Variance", Water Resources Research, Vol 11, No. 2, April 1975, p. 197-204.

ReVelle, C., Joeres, E. and Kirby, W., "The Linear Decision Rule in Reservoir Management and Design, 1. Development of the Stochastic Model", Water Resources Research, Vol. 5, No. 4, 1969, p. 767-777.

ReVelle, C. and Kirby, W., "Linear Decision Rule in Reservoir Management and Design, 2. Performance Optimization", Water Resources Research, Vol. 6, p. 1033-1044, 1970.

Roefs, Theodore G., "Reservoir Management: The State of the Art", I.B.M. Washington Scientific Center, July 15, 1968.

Roefs, T.G. and Bodin, L.D., "Multireservoir Operation Studies", Water Resources Research, Vol. 6, p. 410-420, 1970.

Roefs, T.G. and Guitron, R.A., "Stochastic Reservoir Models: Relative Computational Effort", Water Resources Research, Vol. 11, p. 801-804, 1975.

Rood, Omar E., Dessouky, Mohaded I. and Meredith, Dale P., "Optimal Dynamic Operation of M Serially Linked Reservoirs for N Time Periods", Presented at the 44th National ORSA Meeting, November 1973.

Rosenthal, R.E., "Optimal Reservoir Management: Problems and Models", Presented at the ORSA/TIMS Annual Meeting, Colorado Springs, Colorado, November 10-11-12, 1980.

Rosenthal, R.E., "Scheduling Reservoir Releases for Maximum Hydropower Benefit by Nonlinear Programming on a Network", presented at the ORSA/TIMS Joint National Meeting, San Francisco, 1977.

Salcedo, Daniel E., "Dynamic Economic Simulation of an Irrigation System", University of Texas at Austin, August 1972.

Schweig, Z. and Cole, J.A., "Optimal Control of Linked Reservoirs", Water Resources Research, Vol. 4. p. 479-498, 1968.

Shapiro, S.S. and Wilk, M.B., "An Analysis of Variance Test for Normality (Complete Samples)", Biometrica, Vol. 52, p. 591-611, December 1965.

Sigvaldson, O.T., "A Simulation Model for Operating a Multipurpose Multireservoir System", Water Resources Research, Vol. 12, No. 12, 1976.

Smith, K., "On the Standard Deviations of Adjusted and Interpolated Values of an Observed Polynomial Function and Its Constants and the Guidance They Give Towards a Proper Choice Of the Distribution of Observations", Biometrica 12, p. 1-25, 1918.

Sobel, M.J., "Reservoir Management Models", Water Resources Research, Vol. 11, No. 6, December 1975, p. 767-776.

Su, S.Y. and Deininger, R.A., "Modeling the Regulation of Lake Superior Under Uncertainty of Future Water Supplies", Water Resources Research, Vol. 10. p. 11-25, 1974.

Su, Shraw Y. and Deininger, Rolf A., "Optimal Operation Policies for Multi-Purpose Multi-Reservoir System", 40th National ORSA Meeting, October 25-29, 1971, Anaheim, California.

Takeuchi, K. and Moreau, D.H., "Optimal Control of Multiunit Interbasin Water Resource System", Water Resources Research, Vol. 10, p. 407-414, 1974.

Tennessee Valley Authority Water Resource Management Methods Staff, "Development of a Comprehensive TVA Water Resource Management Program", 1974.

Tennessee Valley Authority Water Resource Management Methods Staff, "Evaluation of Weekly Hydropower Benefits", 1974a.

Texas Water Development Board, Economic Optimization and Simulation Techniques for Management Of Regional Water Resource Systema, TWDB Report 179, 1974a.

Texas Water Development Board, Analytic Techniques for Planning Complex Water Resource Systems, TWDB Report 183, 1974b.

Texas Water Development Board, Water Supply Allocation Model AL-IV, 1975.

Torabi, M. and Mobasheri, F., "A Stochastic Dynamic Programming Model for the Optimum Operation of a Multi-purpose Reservoir", Water Resources Research, Vol. 9, p. 1089-1099, 1973.

Trott, W.J. and Yeh, W.W.G., "Optimization of Multiple Reservoir System", Journal of Hydraulics Division, A.S.C.E., Vol. 99, No. HYRO, p. 1865-1884, 1973.

Weirs, A.O. and Beard, L.R., "A Multibasin Planning Strategy", Texas Water Development Board, 1971, Water Resources Bulletin, Vol. 7, No. 4, August 1971, p. 750-764.

Wilde, D, and Beightler, C., Foundations of Optimization, Prentice Hall, Englewood Cliffs, New Jersey, 1967.

Windsor, James S. and Chow, Ven Te, "Multireservoir Optimization Model", Journal of Hydraulic Division, Proceedings of American Society of Civil Engineers, October 1972, p. 1827-45.

Wismer, D.A. and Challergy, R., Introduction to Nonlinear Programming, North Holland, 1079.

Yeh, W.W.G., "Optimization of Real Time Daily Operation of a Multiple Reservoir System", Report No. UCLA-ENG-7828, Department of Engineering Systems, U.C.L.A., Abstract in Selected Water Resources Abstracts, Vol. 9, No. 23, p. 19, 1976.

Young, G.K., "Finding Reservoir Operating Rules", Proceedings, American Society of Civil Engineers 98 (HY3), 1967, p. 297-321.

VITA

Douglas DeWitt Cochard was born in Kendalville, Indiana, on April 26, 1945, the son of Rex R. Cochard and Ellen R. Cochard. After completing his work at Auburn High School, Auburn, Indiana, in 1963 he entered Bowling Green State University at Bowling Green Ohio. He received the degree of Bachelor of Science with a major in Administrative Science in May 1967. In November of 1967 he entered the United States Air Force. In 1969, he married Christine M. Wimer of Eugene, Oregon. In 1970, he entered the Air Force Instituute of Technology at Wright Patterson AFB, Ohio and received the degree of Master of Science with a major in Systems Management in December 1971. Attending night school while stationed in Florida, he received the degree of Master of Science with a major in Operations Research in 1976 from the Florida Institute of Technology at Melbourne, Florida. In 1977 his wife gave birth to a daughter, Jamie. He currently holds the rank of Major in the USAF.

Permanent Address: RR#4
Auburn, Indiana 46707

This Dissertation was typed by Douglas D. Cochard on an Apple II computer.

END

DATE

FILMED

12-81

DTIC